This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

PRINT OF DRAWINGS
AS ORIGINALLY FILED

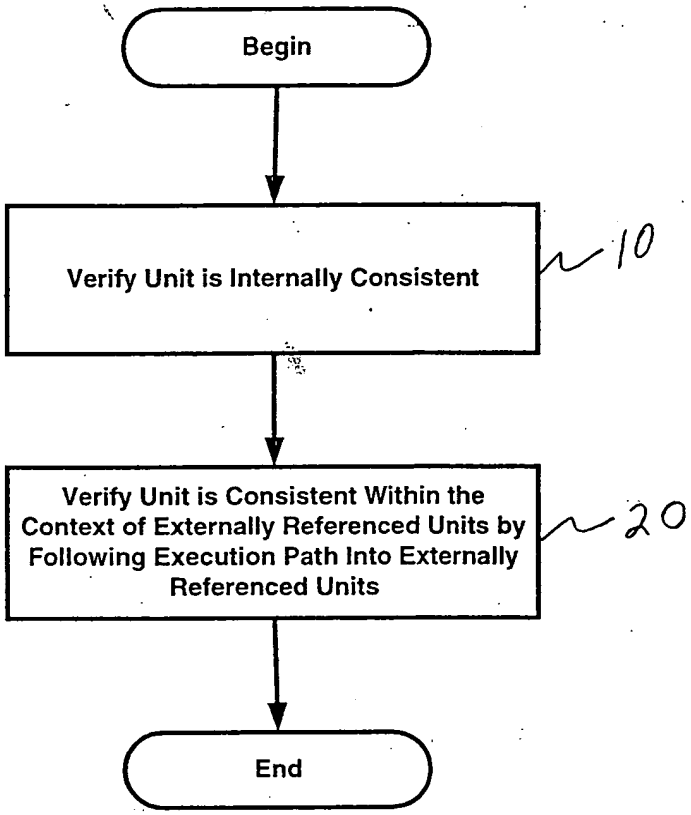

Fig. 1 - Prior Art

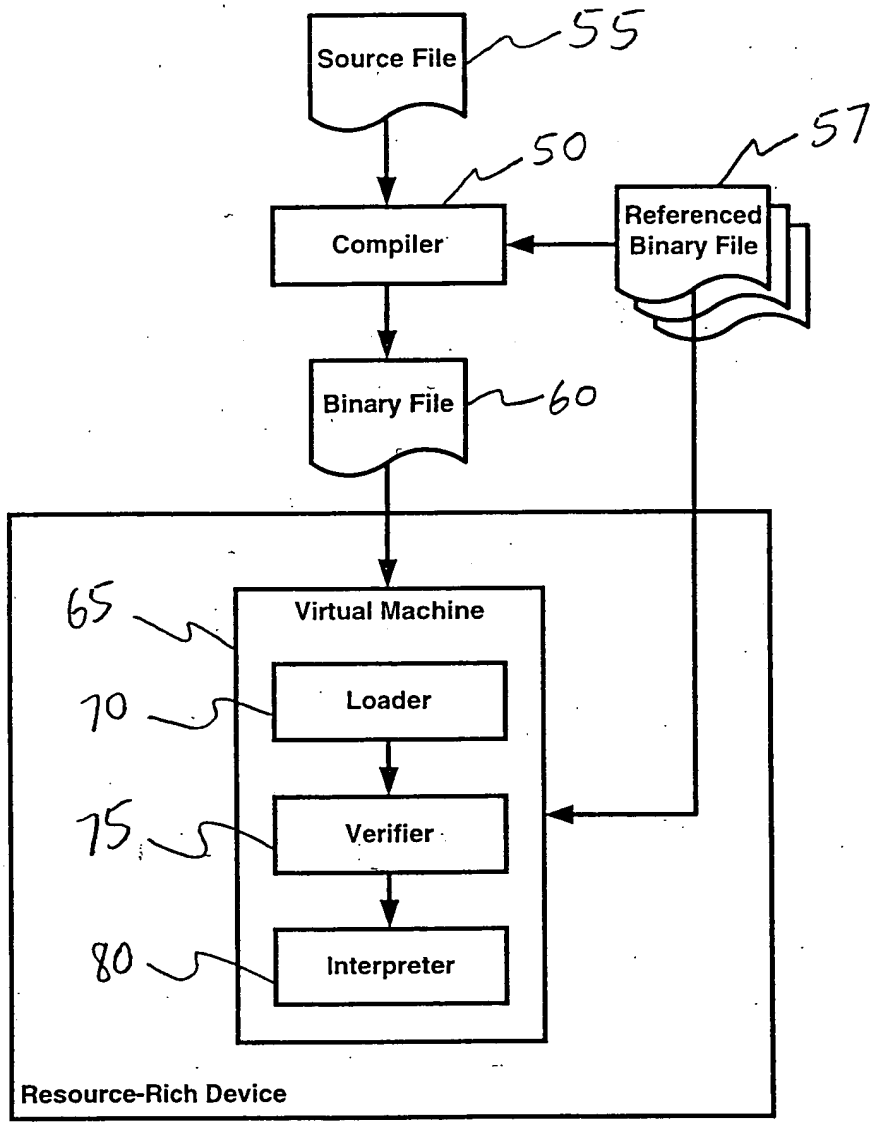

Fig. 2 - Prior Art

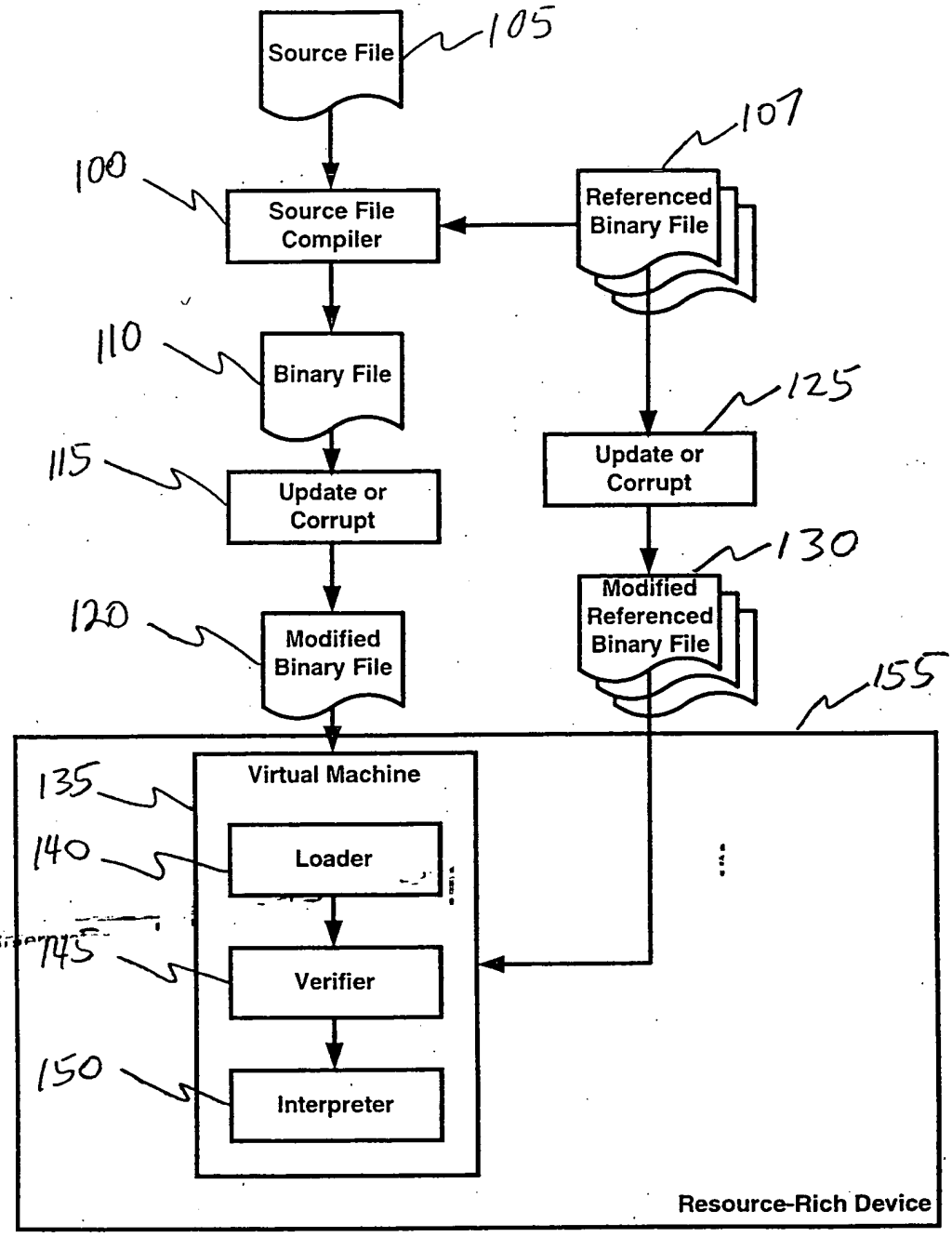

Fig. 3 - Prior Art

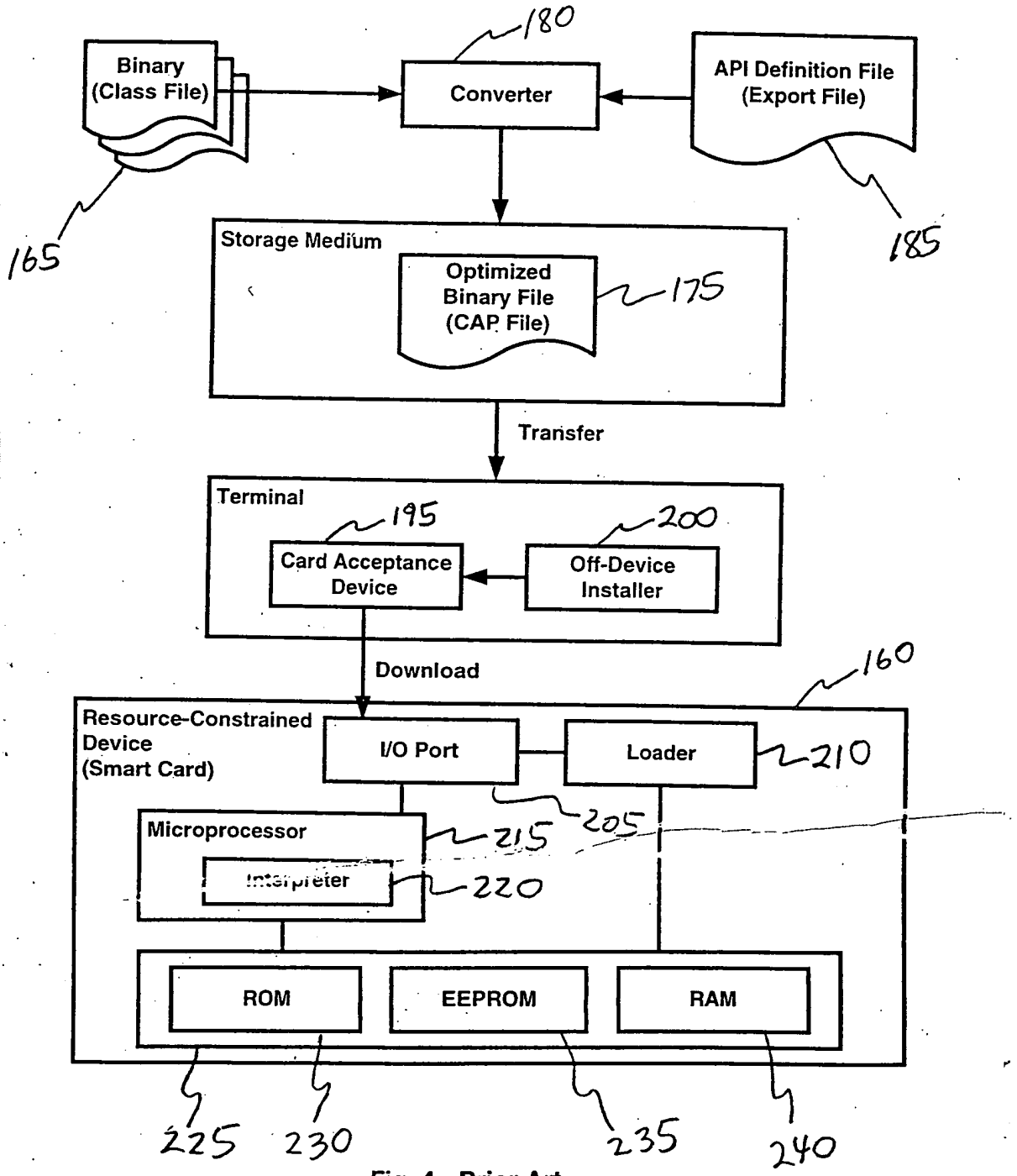

Fig. 4 - Prior Art

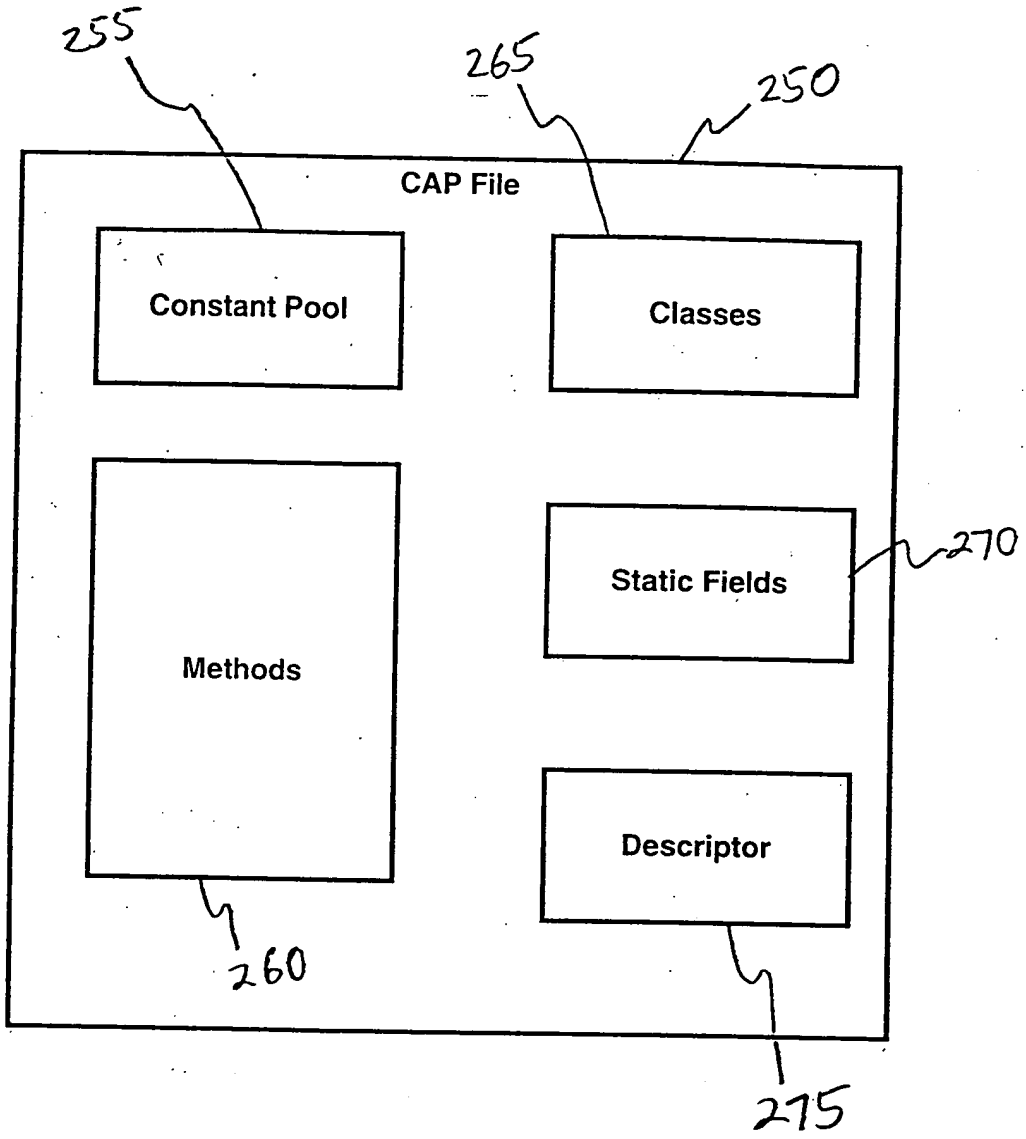

Fig. 5 - Prior Art

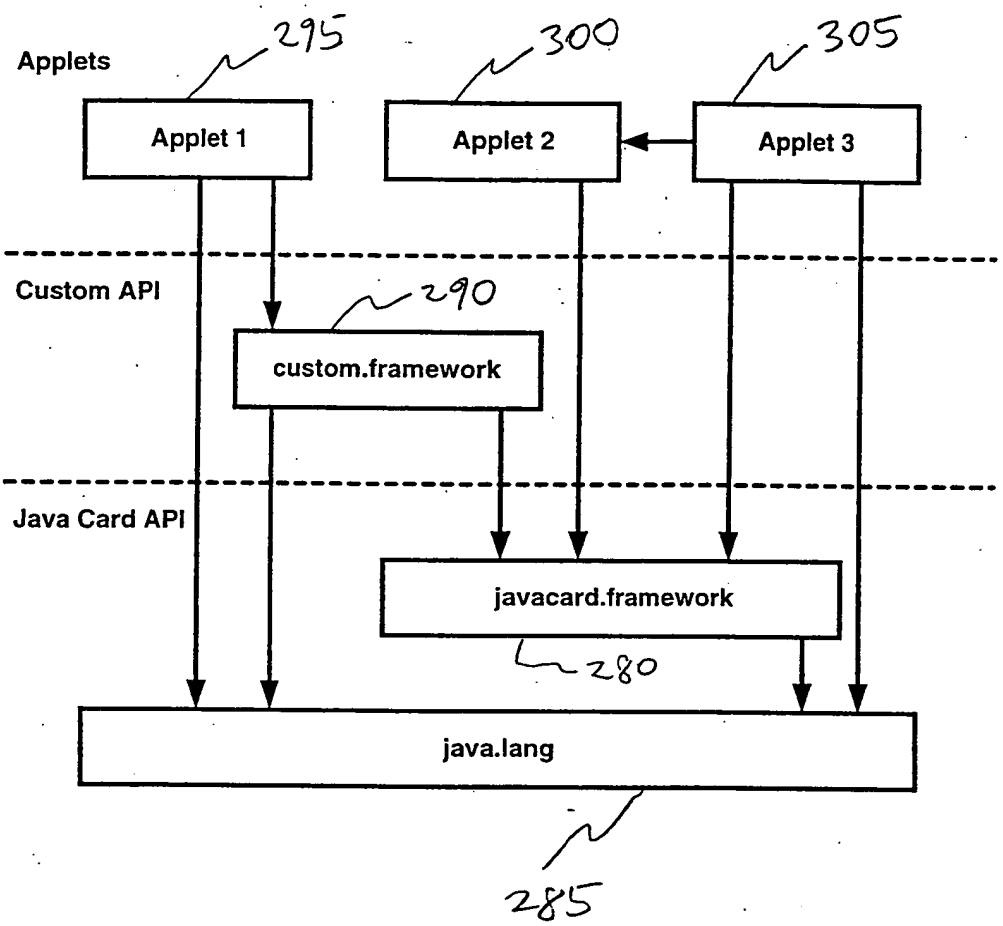

Fig. 6 - Prior Art

PRINT OF DRAWINGS
AS ORIGINALLY FILED

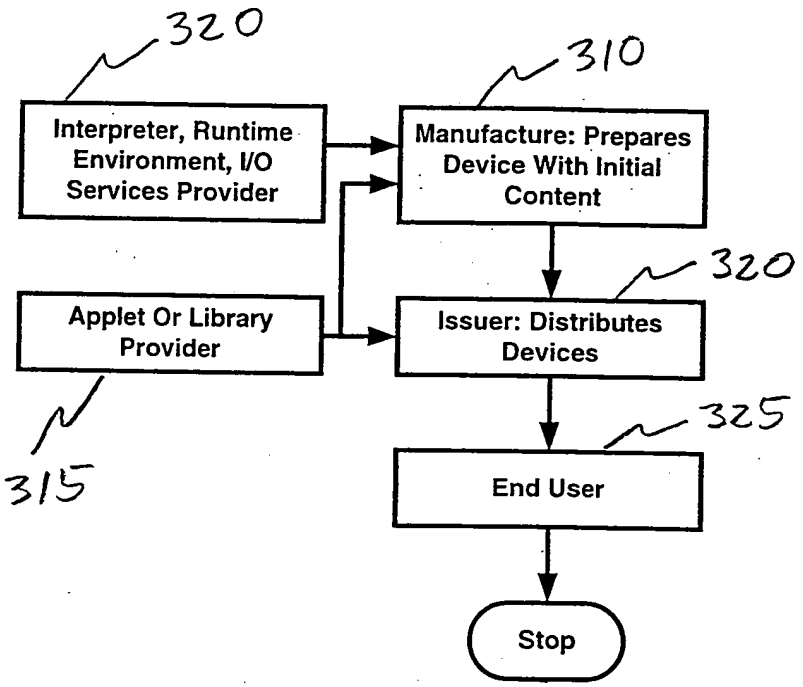

Fig. 7 - Prior Art

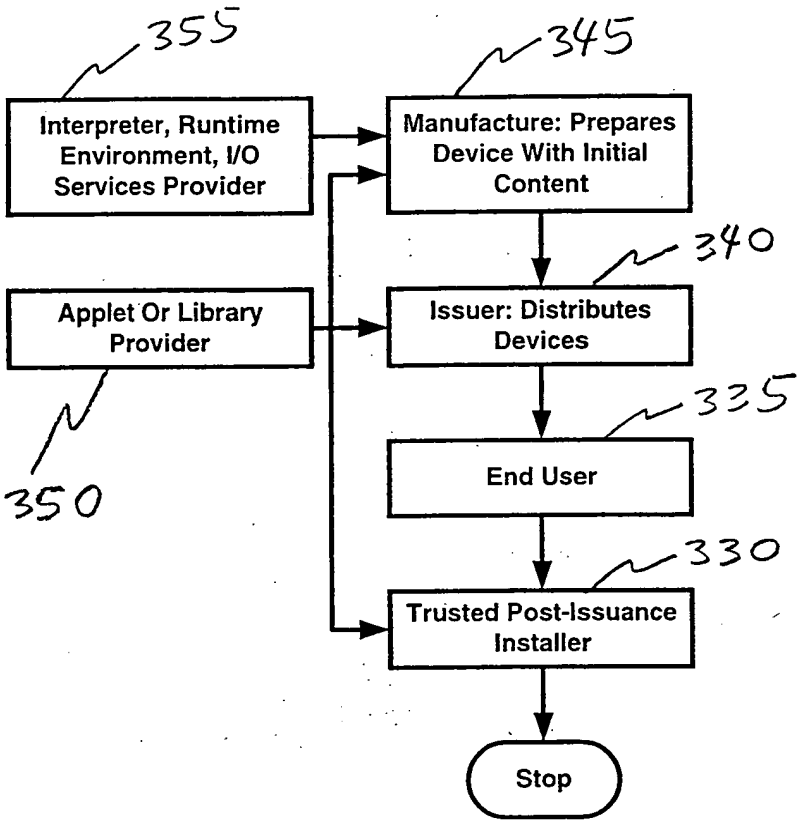

Fig. 8 - Prior Art

PRINT OF DRAWINGS
AS ORIGINALLY FILED

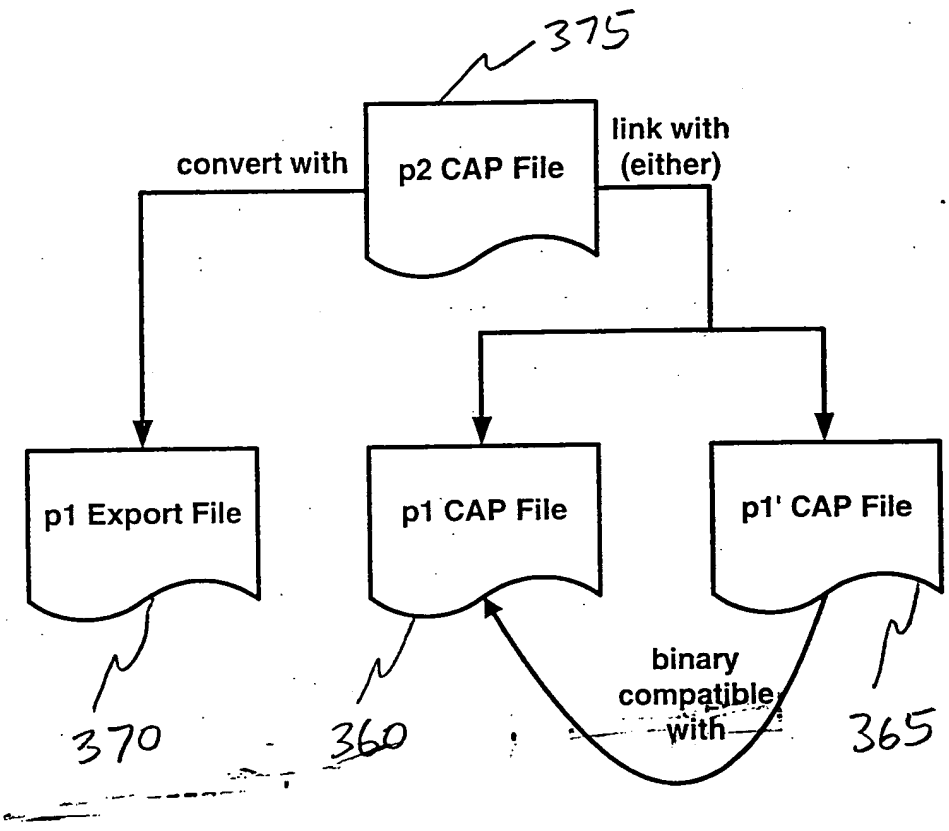

Fig. 9 - Prior Art

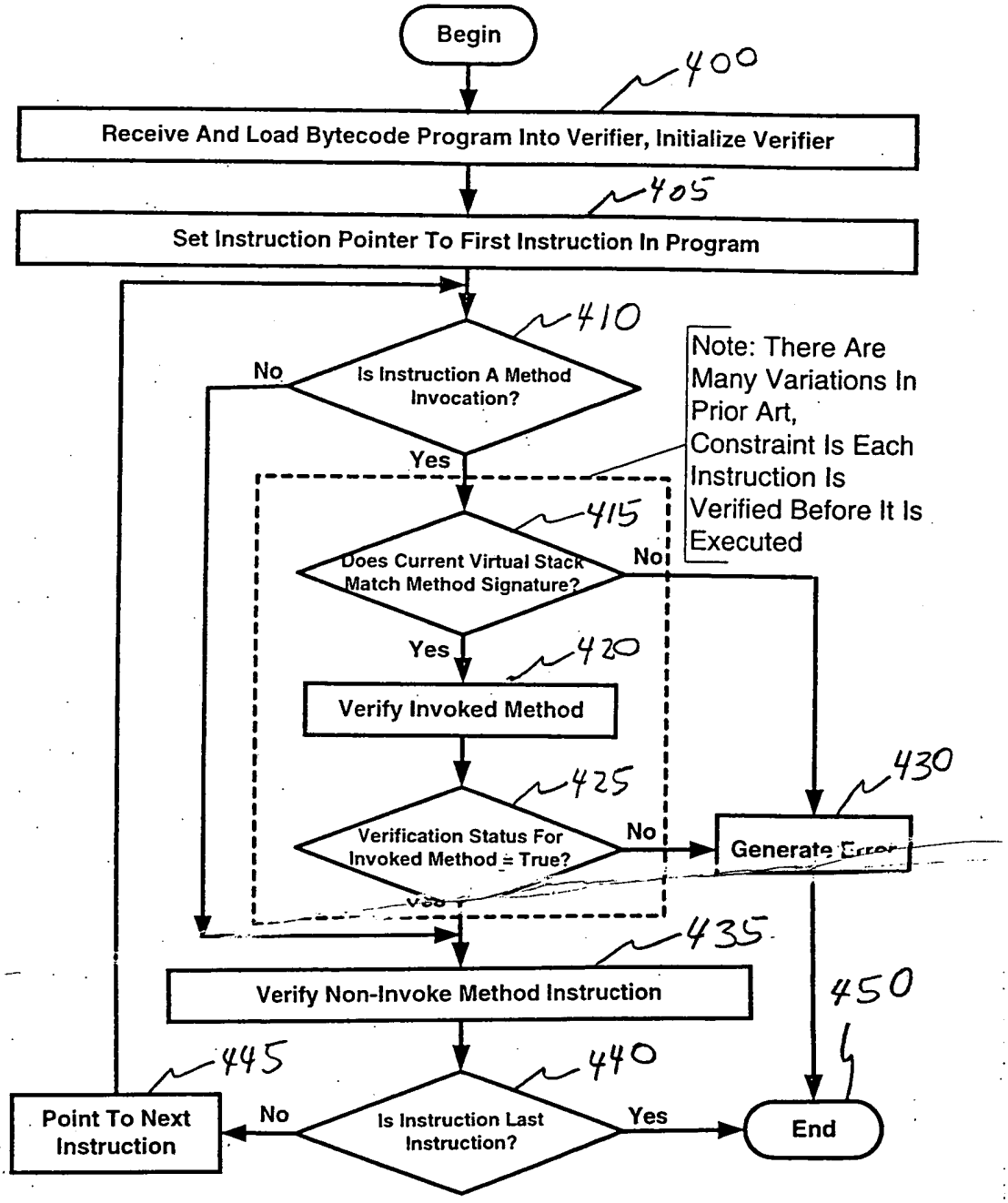

Fig. 10A - Prior Art

PRINT OF DRAWINGS
AS ORIGINALLY FILED


```
package a1;                                   505
import L0;

                                              580
class A1 extends Object {                          555
    void A10 (short s) {                           530
520         byte ba[] = A11(s);
            ...                                    560
            return;
    }
    byte[] A11 (int i) {
525         byte ba[] = new byte[i];                610
            ...
            ba[0] = (byte)classL0.L01(this);
565         ...
            return ba;                             540
    }
    ...
}
```

```
package L0;                                   500

                                                   575
public class classL01 {
    public static int L01 (Object o) {
535         int i;                                 595
            float f = L02(i,o);
585         ...                                    550
            return i;
    }
    private static float L02 (int i,Object o) {
545         float f;                               605
            ...
            return f;                              600
    }
    ...
}
```

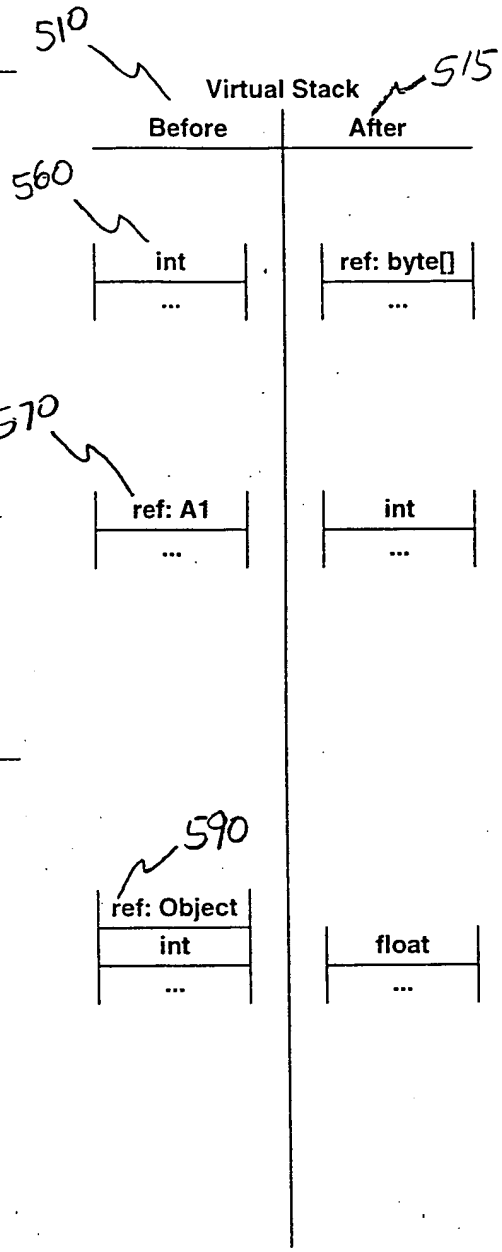

Fig. 10B - Prior Art

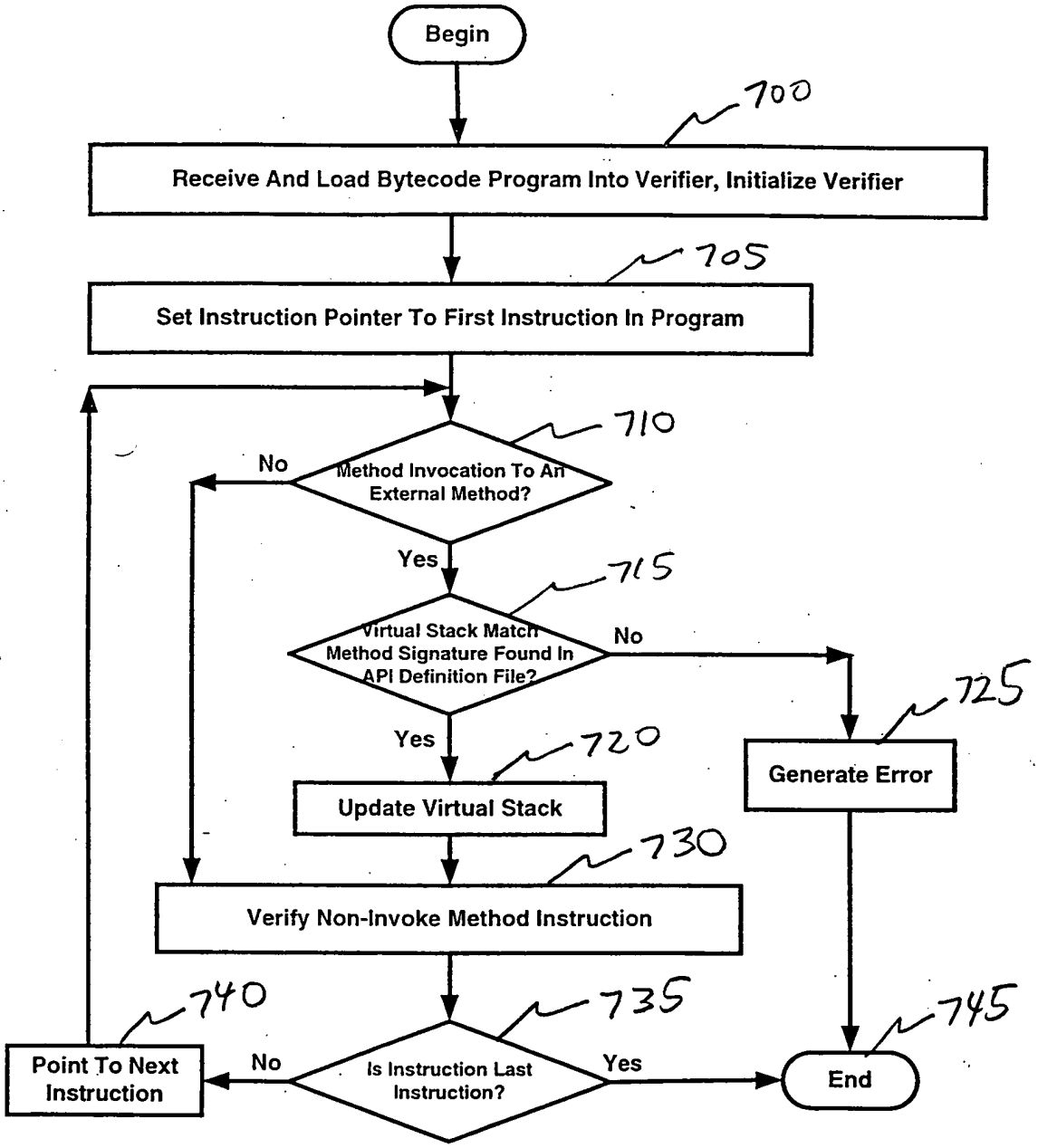
Begin
700
Receive And Load Bytecode Program Into Verifier, Initialize Verifier
705
Set Instruction Pointer To First Instruction In Program
710
Method Invocation To An External Method?
No
Yes
715
Virtual Stack Match Method Signature Found In API Definition File?
No
Yes
720
Update Virtual Stack
725
Generate Error
730
Verify Non-Invoke Method Instruction
735
Is Instruction Last Instruction?
No
Yes
740
Point To Next Instruction
745
End

Fig. 10C

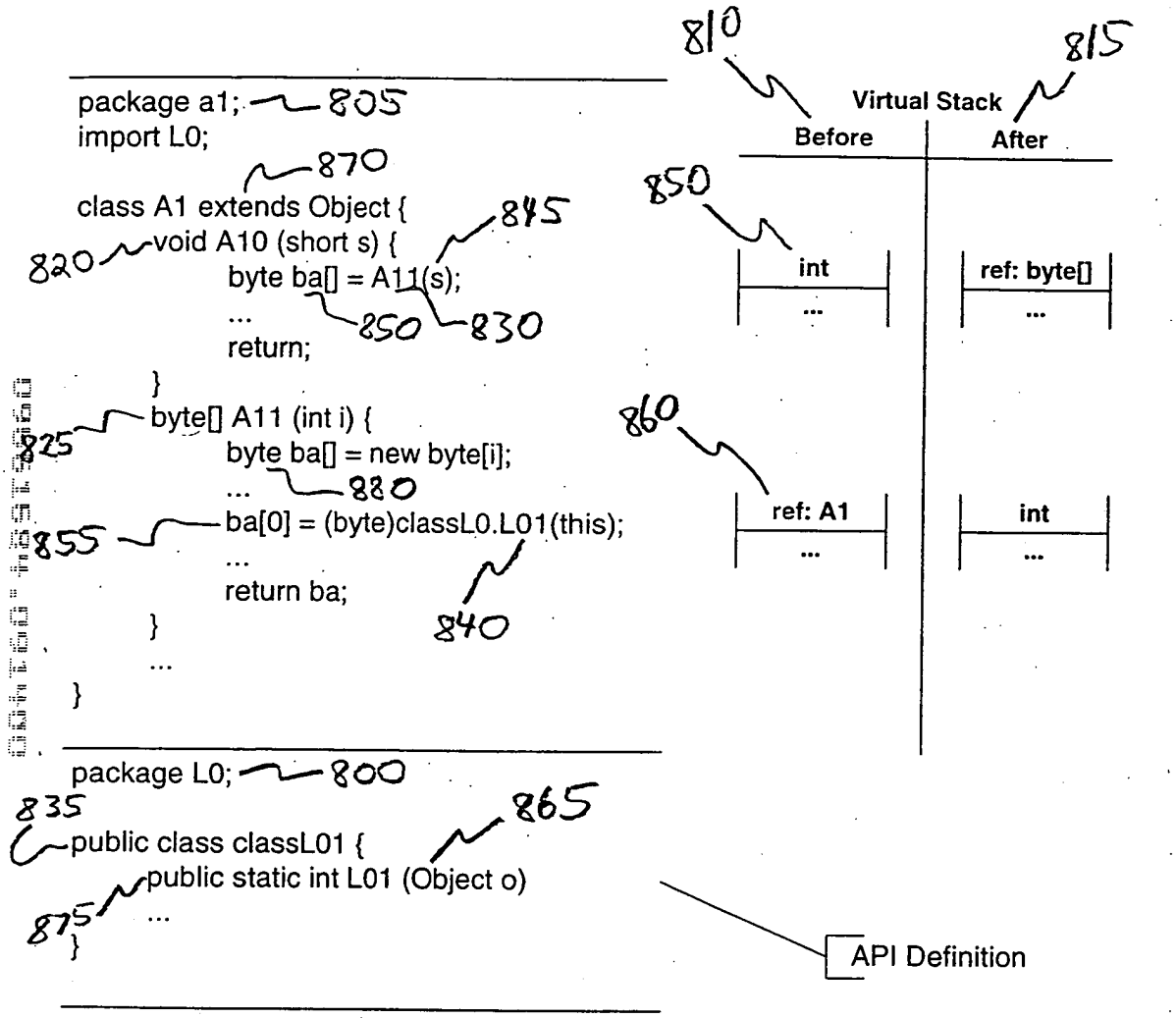
package a1; 805
import L0;
870
class A1 extends Object {
820 void A10 (short s) {
845
byte ba[] = A11(s);
...
850
830
return;
}
825 byte[] A11 (int i) {
byte ba[] = new byte[i];
...
880
855 ba[0] = (byte)classL0.L01(this);
...
return ba;
840
}
...
}
package L0; 800
835
865
public class classL01 {
public static int L01 (Object o)
875 ...
}
810
815
Virtual Stack
Before
After
850
int
...
ref: byte[]
...
860
ref: A1
...
int
...
API Definition

Fig. 10D

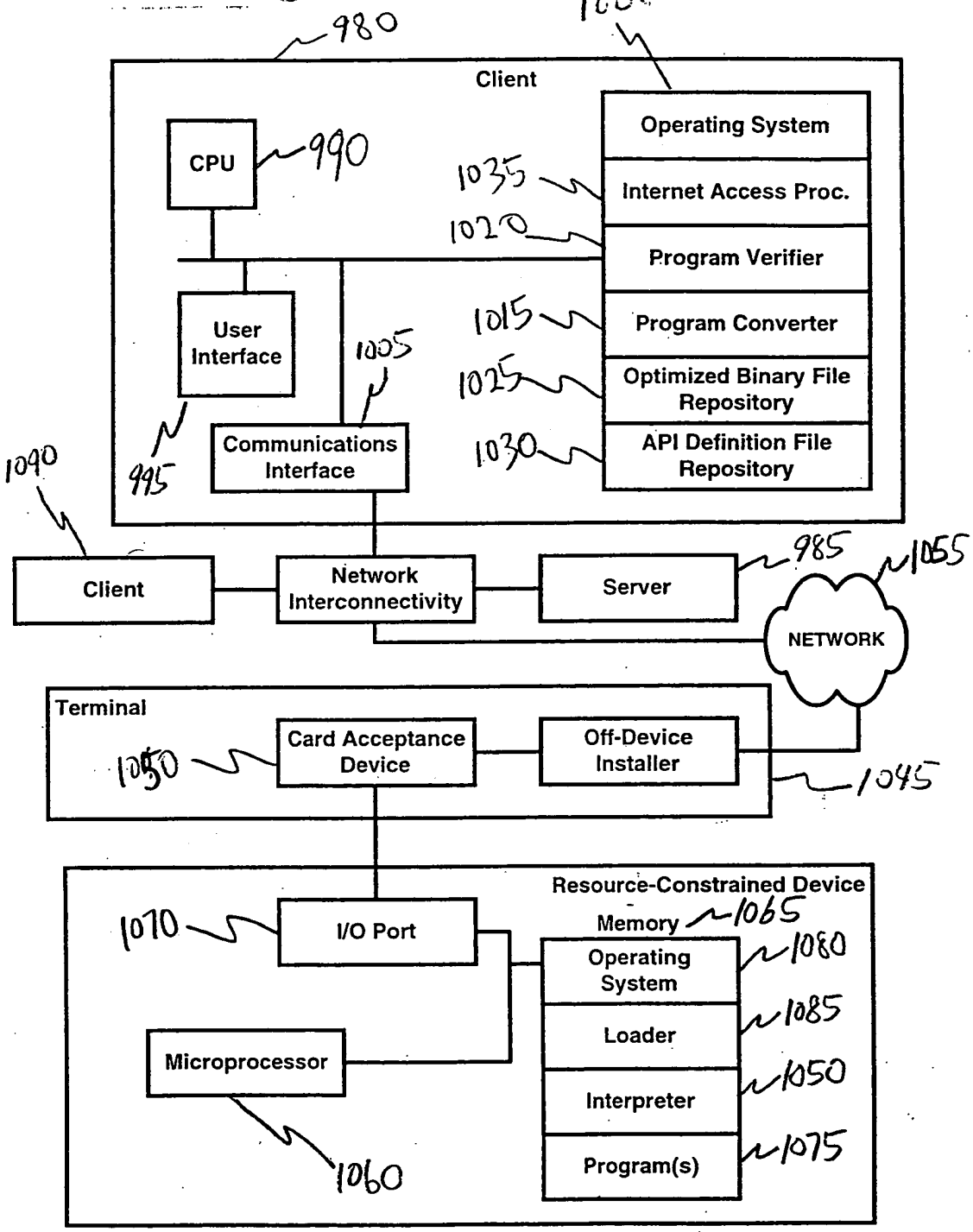
980
100
Client
Operating System
CPU
990
1035
Internet Access Proc.
1020
Program Verifier
User Interface
1015
Program Converter
1005
1025
Optimized Binary File Repository
995
Communications Interface
1030
API Definition File Repository
1090
Client
Network Interconnectivity
Server
985
1055
NETWORK
Terminal
1050
Card Acceptance Device
Off-Device Installer
1045
Resource-Constrained Device
1070
I/O Port
Memory
1065
Operating System
1080
Loader
1085
Microprocessor
Interpreter
1050
Program(s)
1075
1060

Fig. 11A

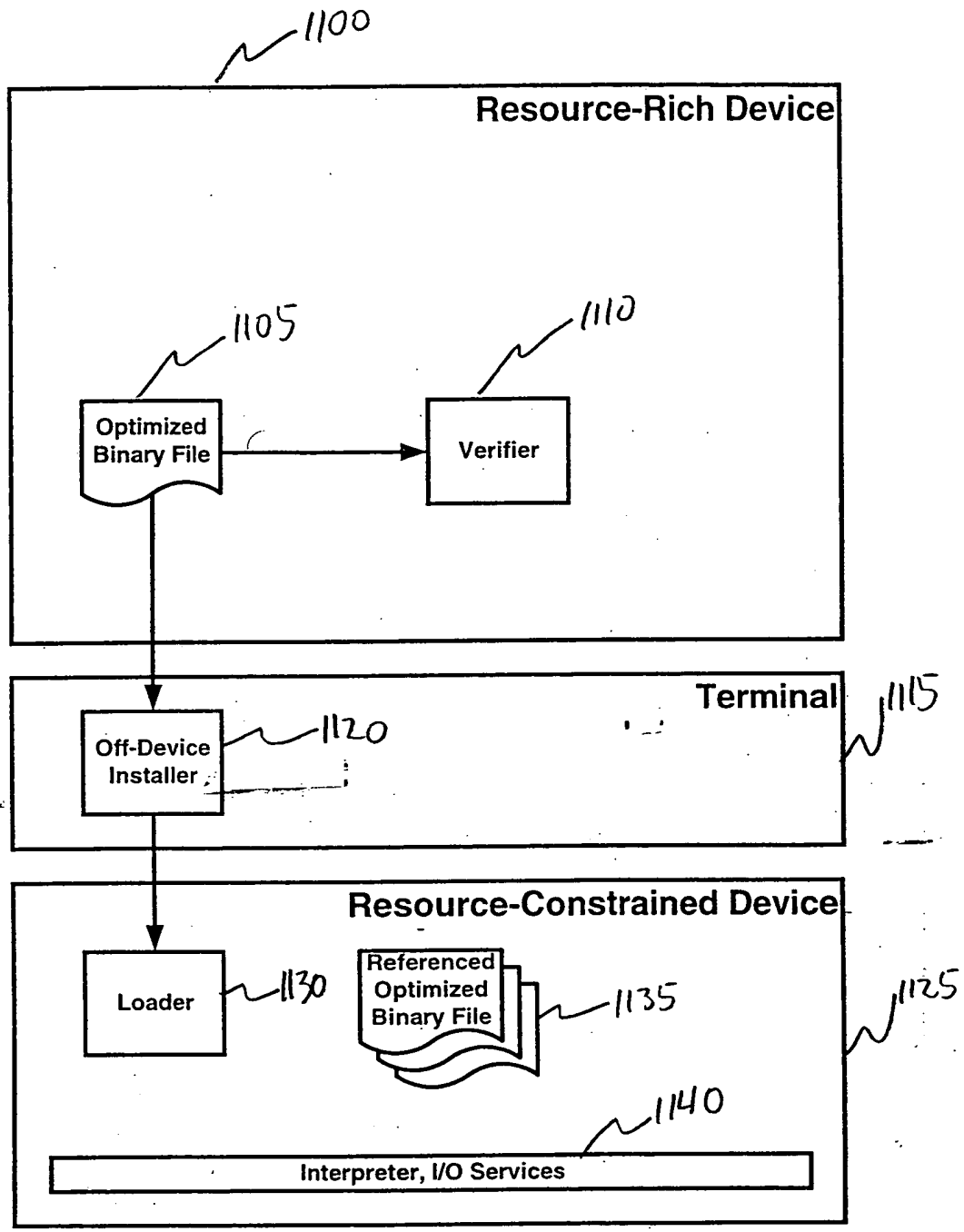
1100
Resource-Rich Device
1105
Optimized Binary File
1110
Verifier
Terminal
1115
Off-Device Installer
1120
Resource-Constrained Device
Loader
1130
Referenced Optimized Binary File
1135
1125
1140
Interpreter, I/O Services

Fig. 11B

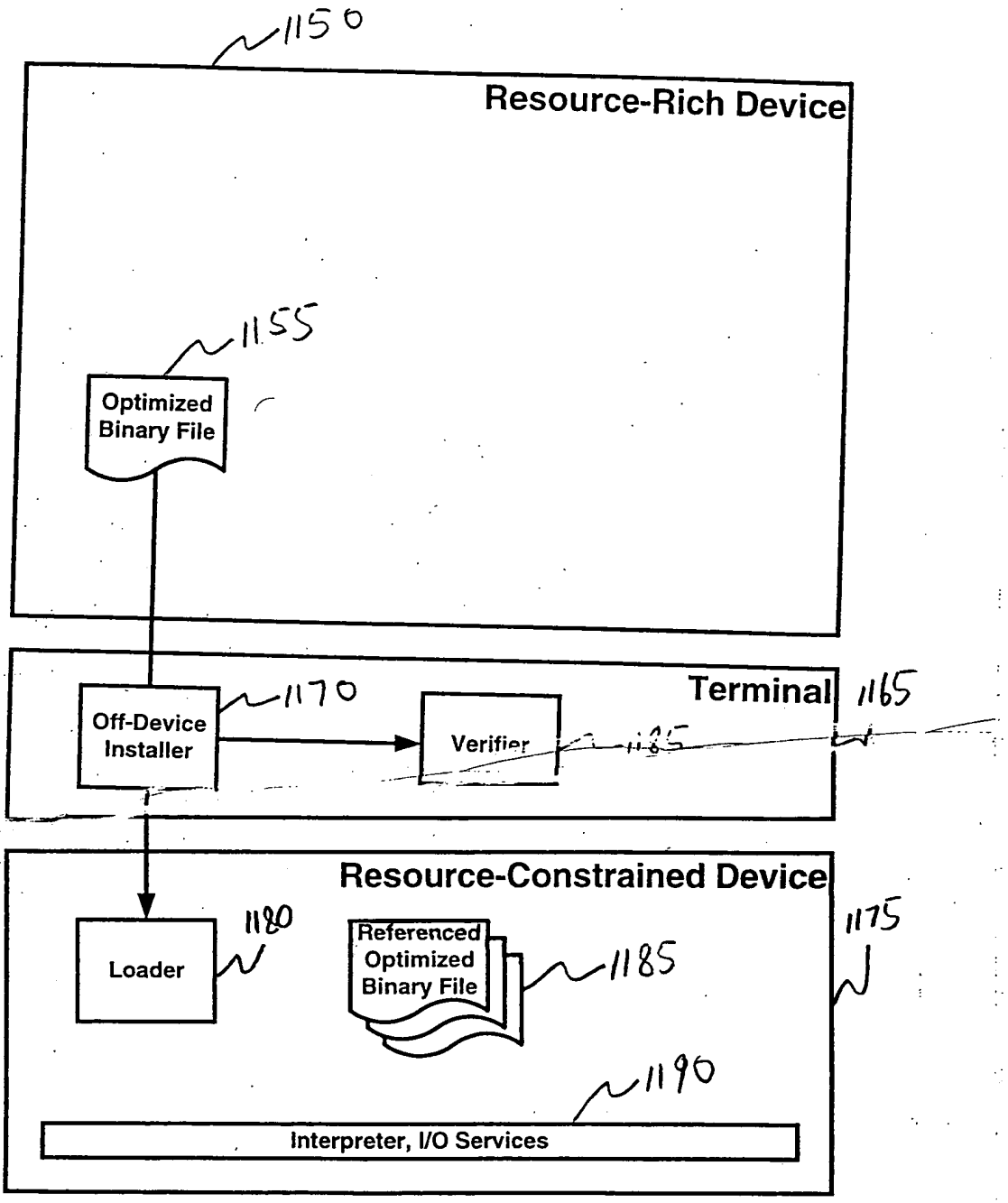
1150
Resource-Rich Device
1155
Optimized Binary File
Terminal
1165
1170
Off-Device Installer
Verifier
1175
Resource-Constrained Device
1180
Loader
Referenced Optimized Binary File
1185
1190
Interpreter, I/O Services

Fig. 11C

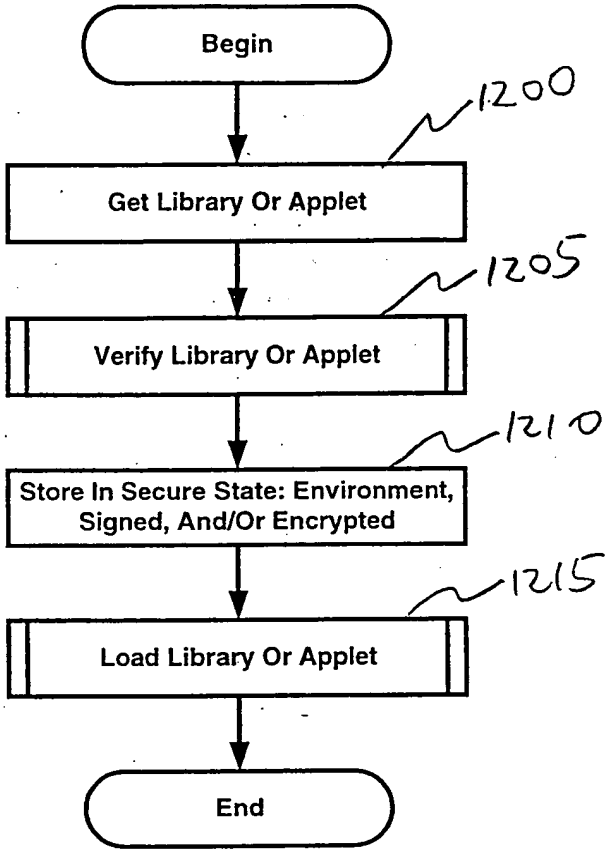
Begin
1200
Get Library Or Applet
1205
Verify Library Or Applet
1210
Store In Secure State: Environment, Signed, And/Or Encrypted
1215
Load Library Or Applet
End

Fig. 12

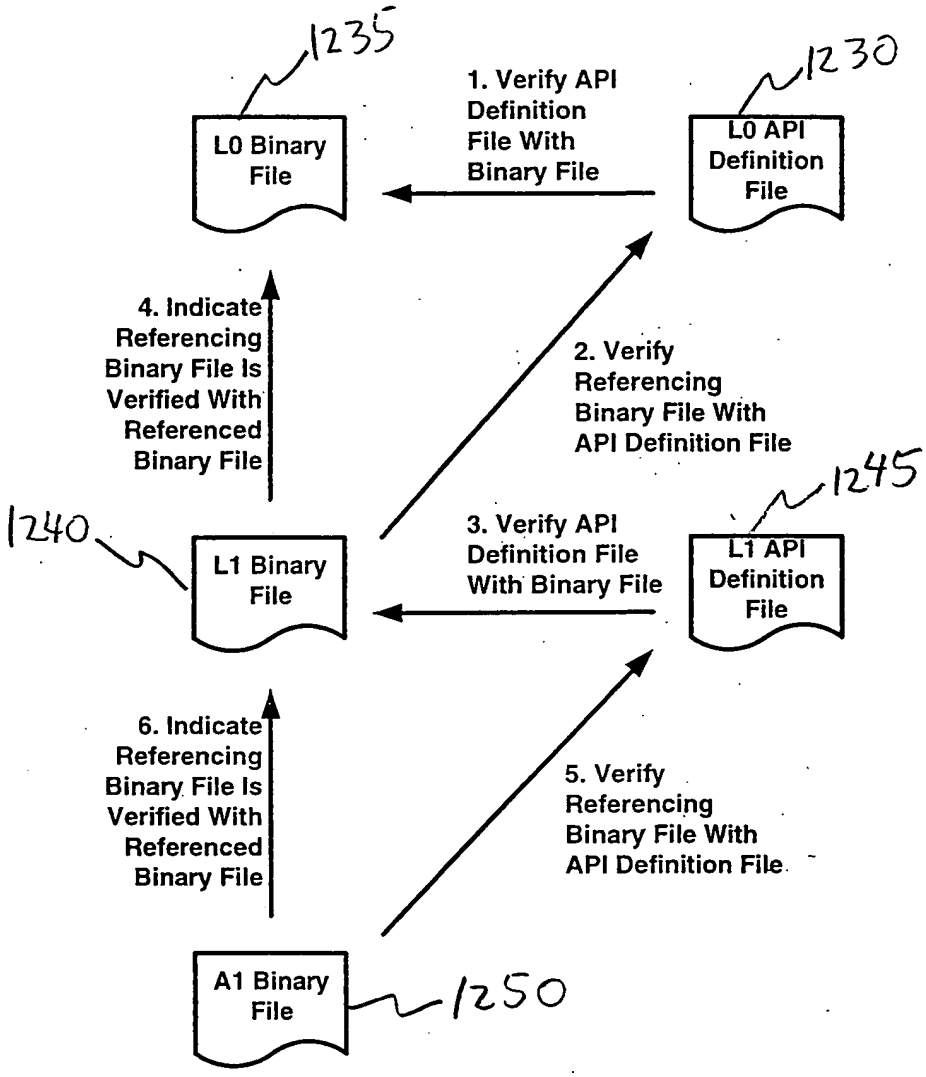
1235
1230
L0 Binary File
1. Verify API Definition File With Binary File
L0 API Definition File
4. Indicate Referencing Binary File Is Verified With Referenced Binary File
2. Verify Referencing Binary File With API Definition File
1245
1240
L1 Binary File
3. Verify API Definition File With Binary File
L1 API Definition File
6. Indicate Referencing Binary File Is Verified With Referenced Binary File
5. Verify Referencing Binary File With API Definition File
A1 Binary File
1250

Fig. 13A

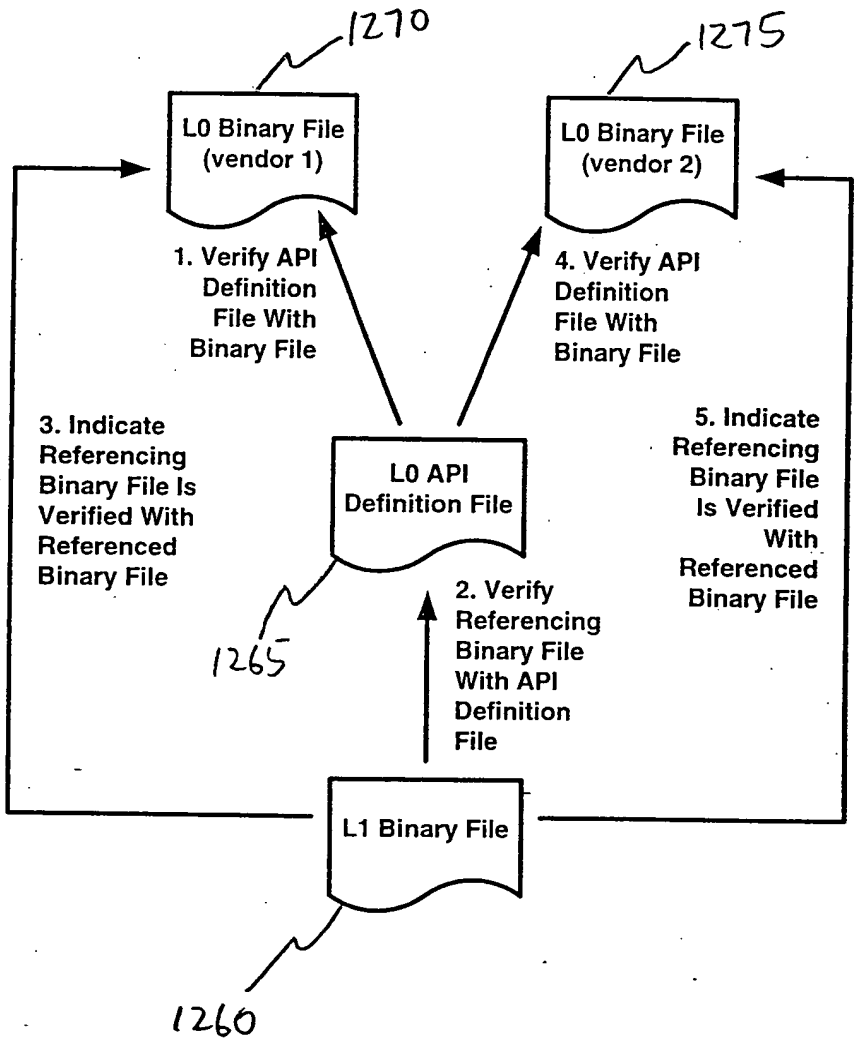
1270
1275
L0 Binary File (vendor 1)
L0 Binary File (vendor 2)
1. Verify API Definition File With Binary File
4. Verify API Definition File With Binary File
3. Indicate Referencing Binary File Is Verified With Referenced Binary File
5. Indicate Referencing Binary File Is Verified With Referenced Binary File
L0 API Definition File
1265
2. Verify Referencing Binary File With API Definition File
L1 Binary File
1260

Fig. 13B

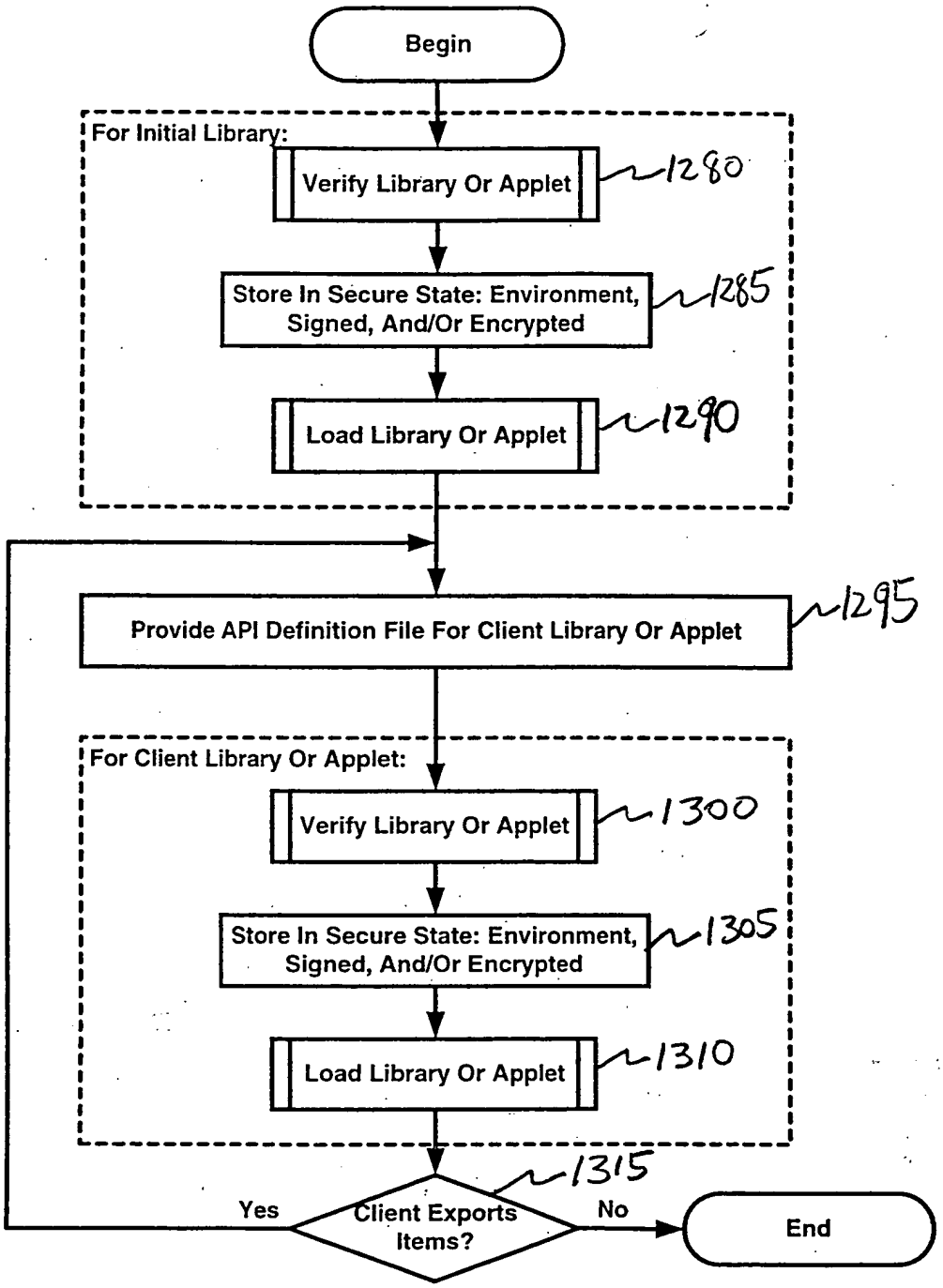
Begin
For Initial Library:
Verify Library Or Applet
1280
Store In Secure State: Environment, Signed, And/Or Encrypted
1285
Load Library Or Applet
1290
Provide API Definition File For Client Library Or Applet
1295
For Client Library Or Applet:
Verify Library Or Applet
1300
Store In Secure State: Environment, Signed, And/Or Encrypted
1305
Load Library Or Applet
1310
Client Exports Items?
1315
Yes
No
End

Fig. 14

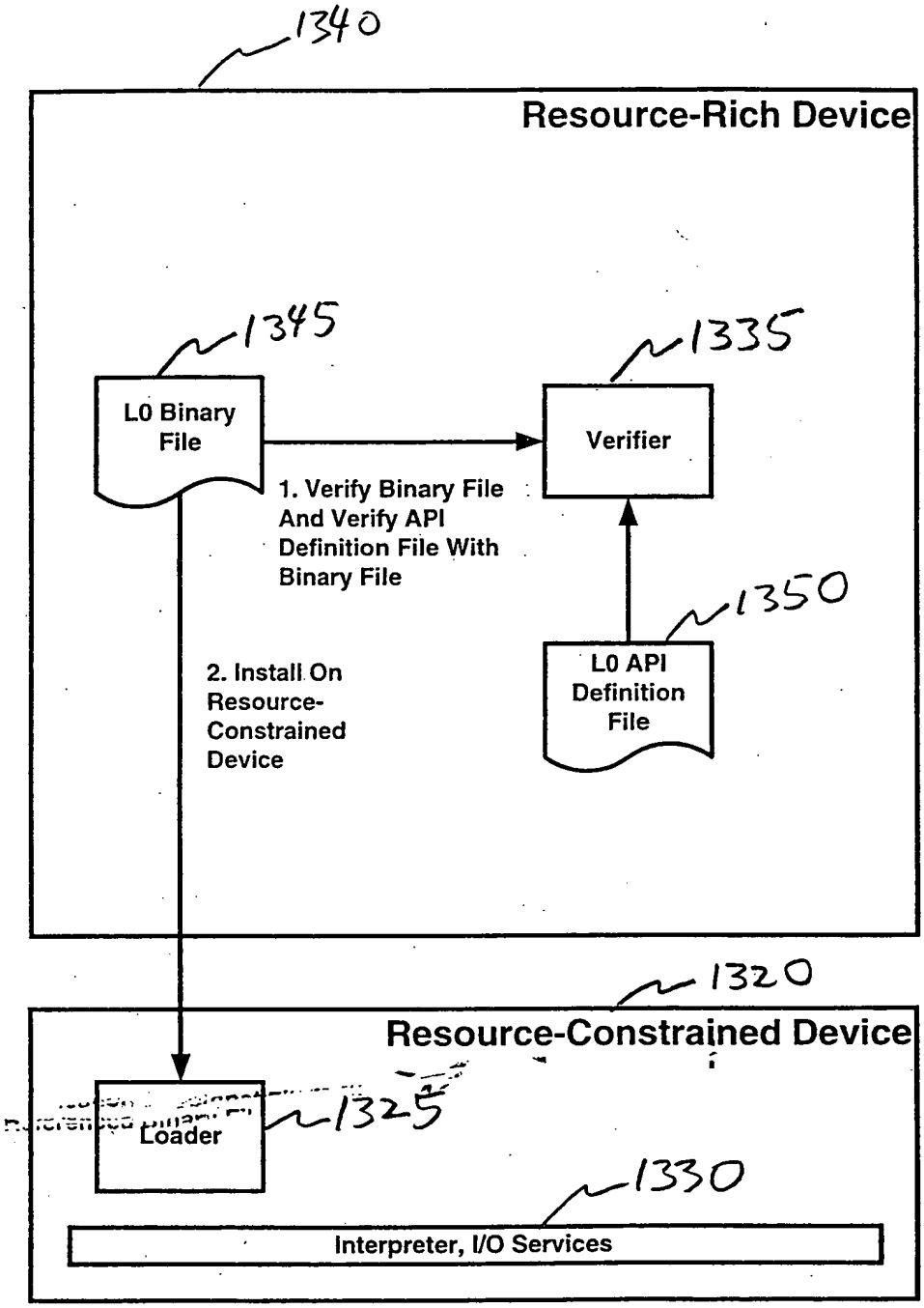
1340
Resource-Rich Device
1345
L0 Binary File
1335
Verifier
1. Verify Binary File And Verify API Definition File With Binary File
1350
L0 API Definition File
2. Install On Resource-Constrained Device
1320
Resource-Constrained Device
Loader
1325
1330
Interpreter, I/O Services

Fig. 15A

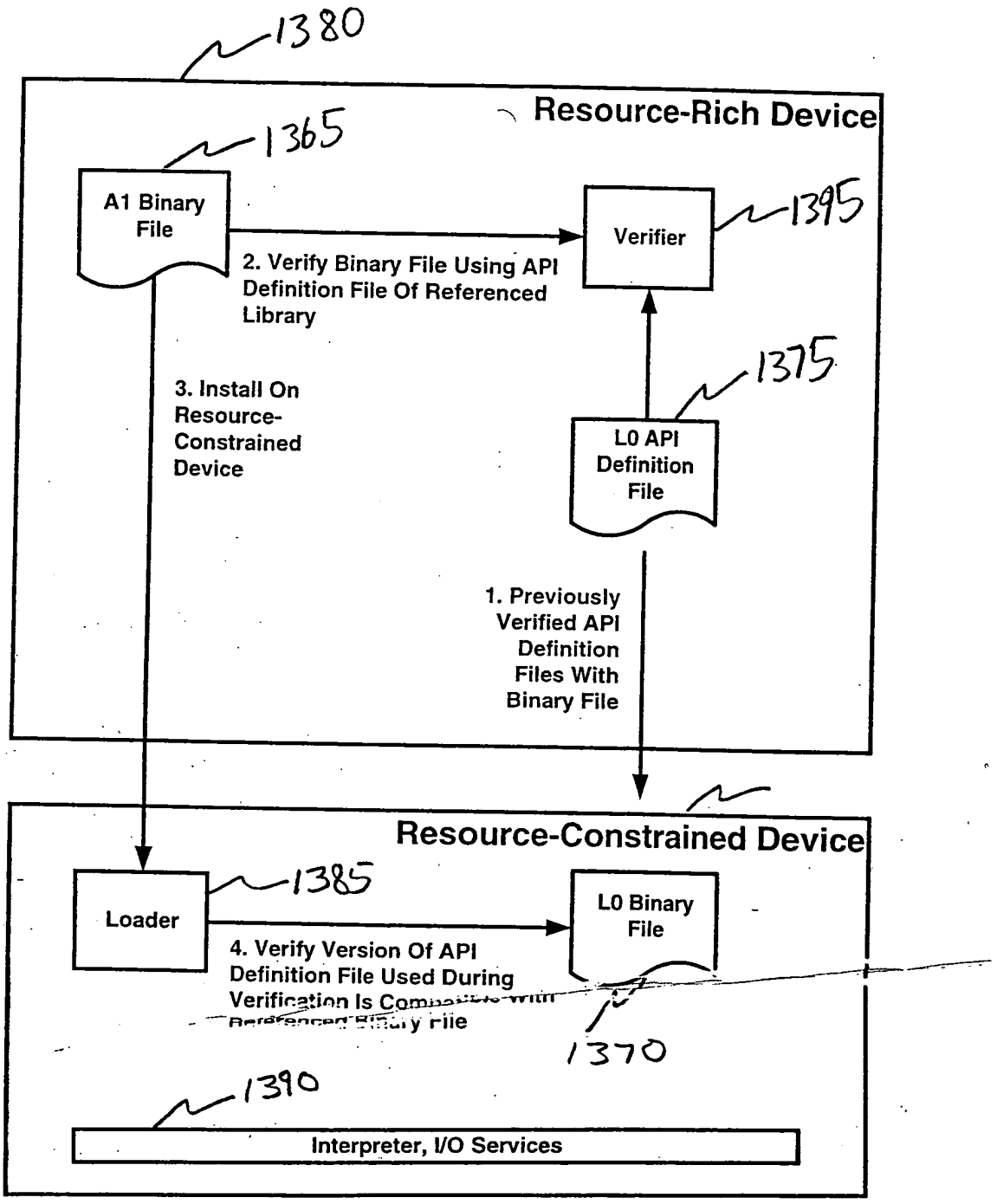
1380
Resource-Rich Device
1365
A1 Binary File
Verifier
1395
2. Verify Binary File Using API Definition File Of Referenced Library
3. Install On Resource-Constrained Device
1375
L0 API Definition File
1. Previously Verified API Definition Files With Binary File
Resource-Constrained Device
Loader
1385
L0 Binary File
4. Verify Version Of API Definition File Used During Verification Is Compatible With Referenced Binary File
1370
1390
Interpreter, I/O Services

Fig. 15B

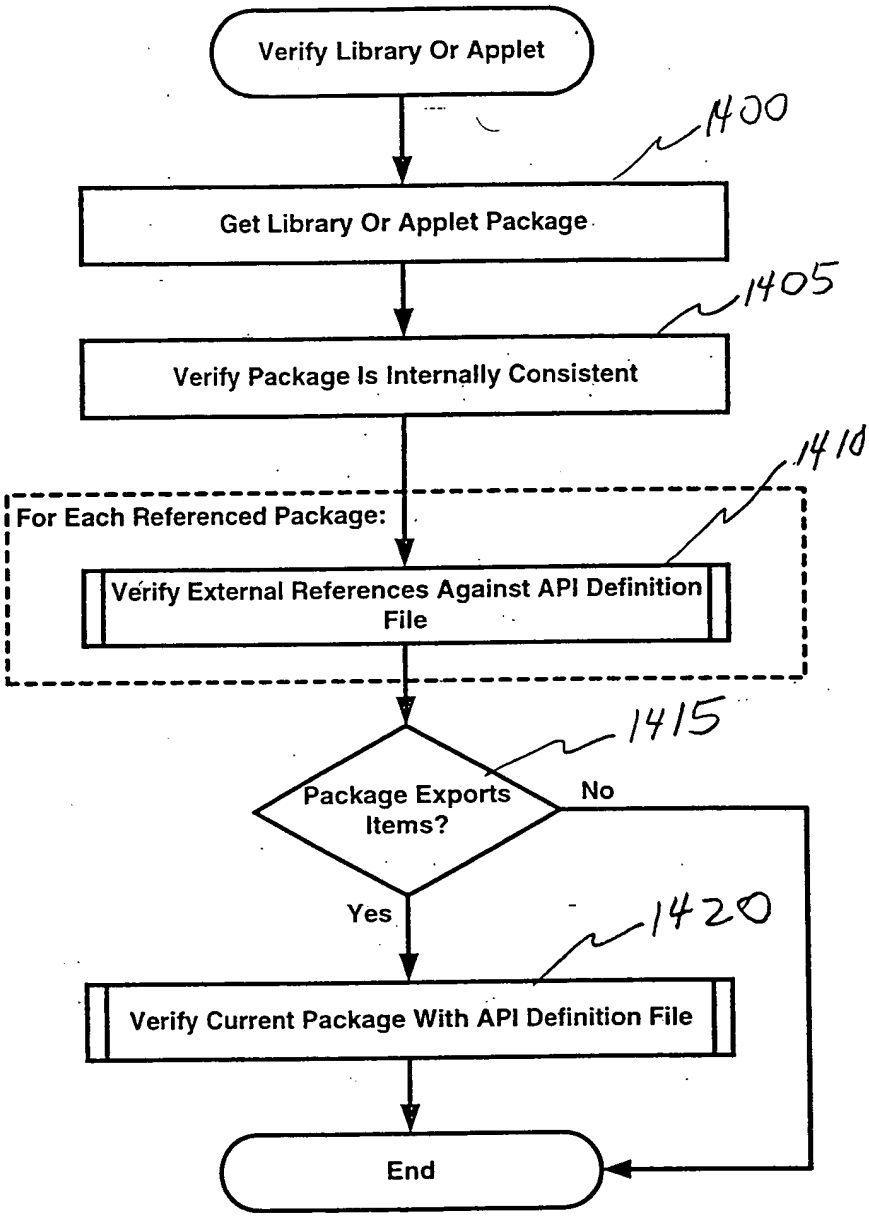
Verify Library Or Applet
1400
Get Library Or Applet Package
1405
Verify Package Is Internally Consistent
1410
For Each Referenced Package:
Verify External References Against API Definition File
1415
Package Exports Items?
No
Yes
1420
Verify Current Package With API Definition File
End

Fig. 16

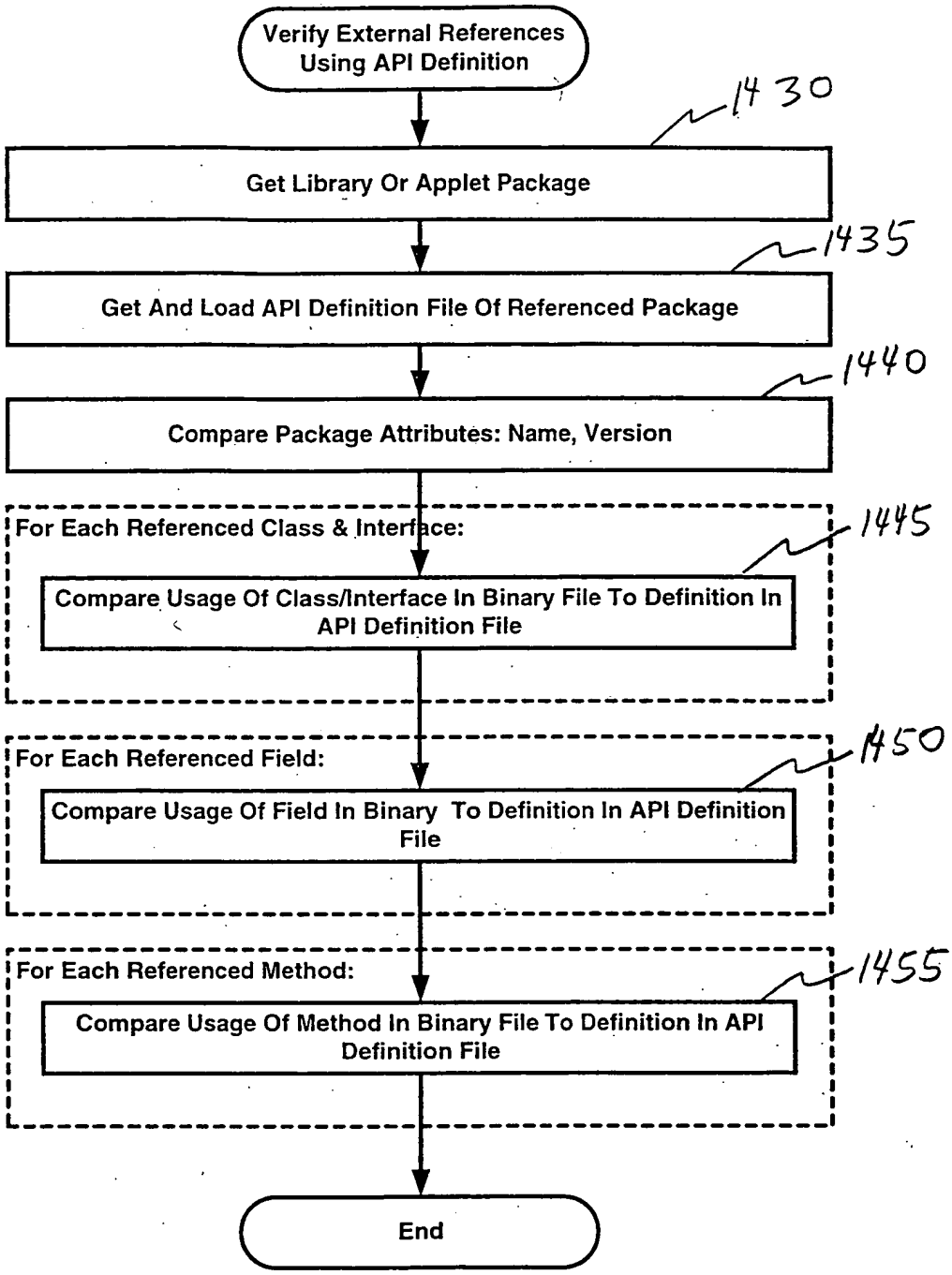
Verify External References Using API Definition
1430
Get Library Or Applet Package
1435
Get And Load API Definition File Of Referenced Package
1440
Compare Package Attributes: Name, Version
For Each Referenced Class & Interface:
1445
Compare Usage Of Class/Interface In Binary File To Definition In API Definition File
For Each Referenced Field:
1450
Compare Usage Of Field In Binary To Definition In API Definition File
For Each Referenced Method:
1455
Compare Usage Of Method In Binary File To Definition In API Definition File
End

Fig. 17

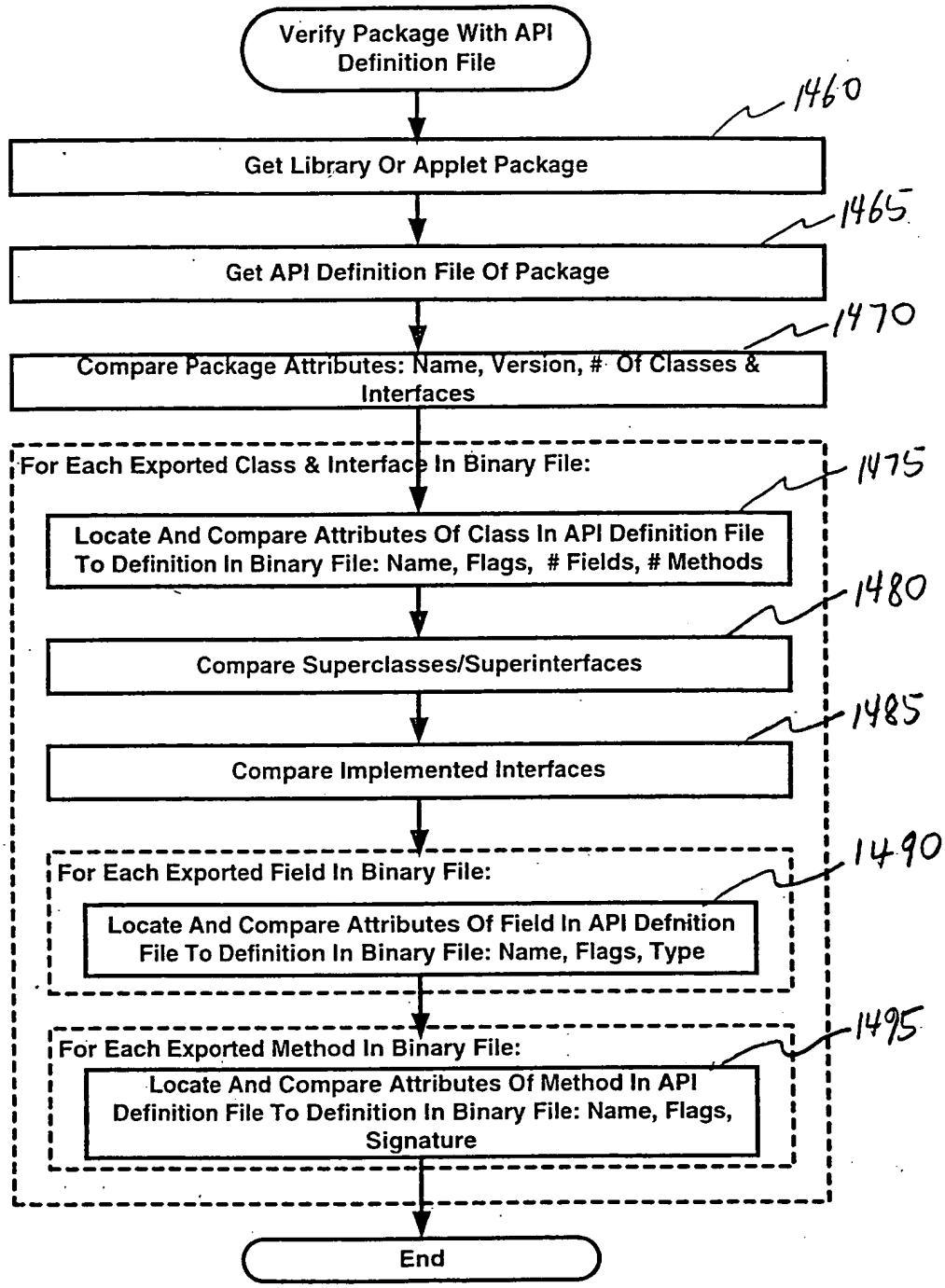
Verify Package With API Definition File
1460
Get Library Or Applet Package
1465
Get API Definition File Of Package
1470
Compare Package Attributes: Name, Version, # Of Classes & Interfaces
For Each Exported Class & Interface In Binary File:
1475
Locate And Compare Attributes Of Class In API Definition File To Definition In Binary File: Name, Flags, # Fields, # Methods
1480
Compare Superclasses/Superinterfaces
1485
Compare Implemented Interfaces
For Each Exported Field In Binary File:
1490
Locate And Compare Attributes Of Field In API Defnition File To Definition In Binary File: Name, Flags, Type
For Each Exported Method In Binary File:
1495
Locate And Compare Attributes Of Method In API Definition File To Definition In Binary File: Name, Flags, Signature
End

Fig. 18

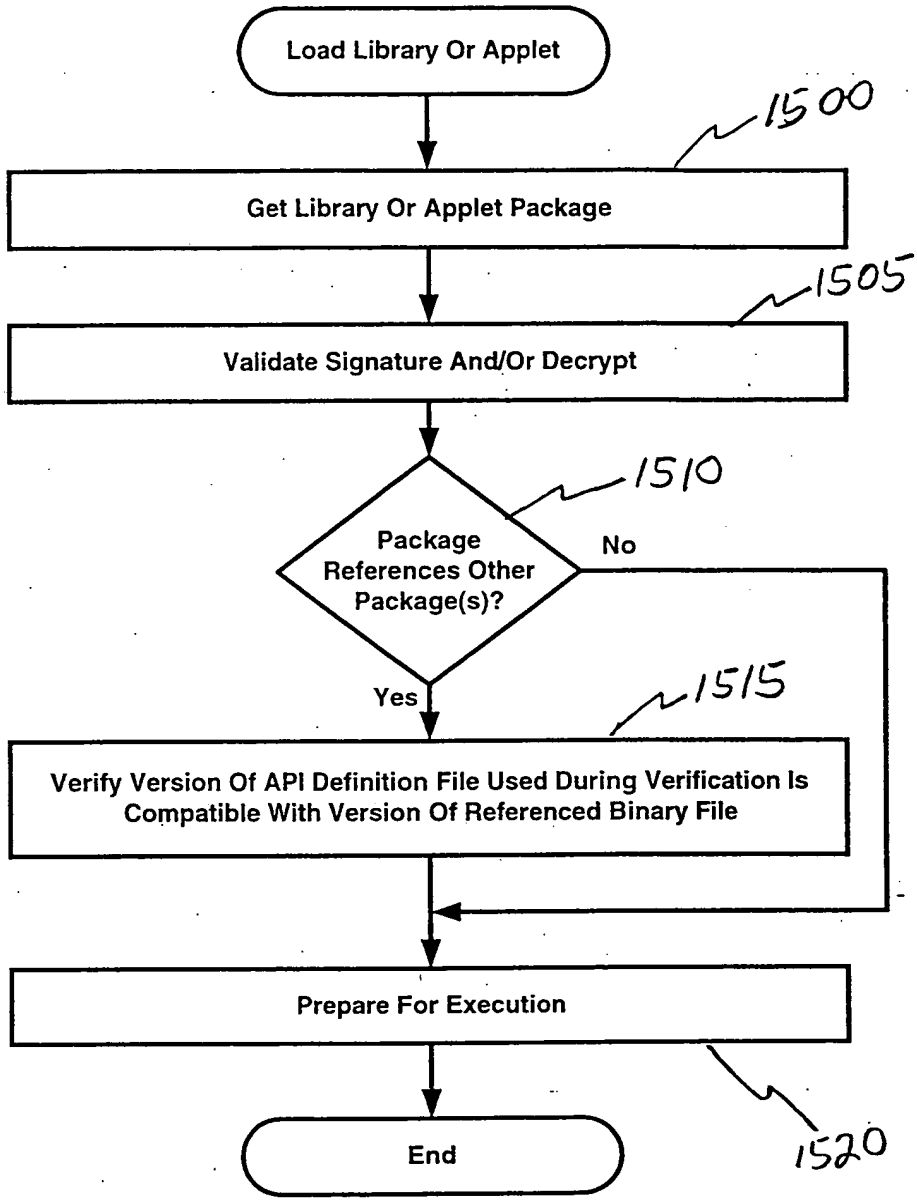
Load Library Or Applet
1500
Get Library Or Applet Package
1505
Validate Signature And/Or Decrypt
1510
Package References Other Package(s)?
No
Yes
1515
Verify Version Of API Definition File Used During Verification Is Compatible With Version Of Referenced Binary File
Prepare For Execution
End
1520

Fig. 19

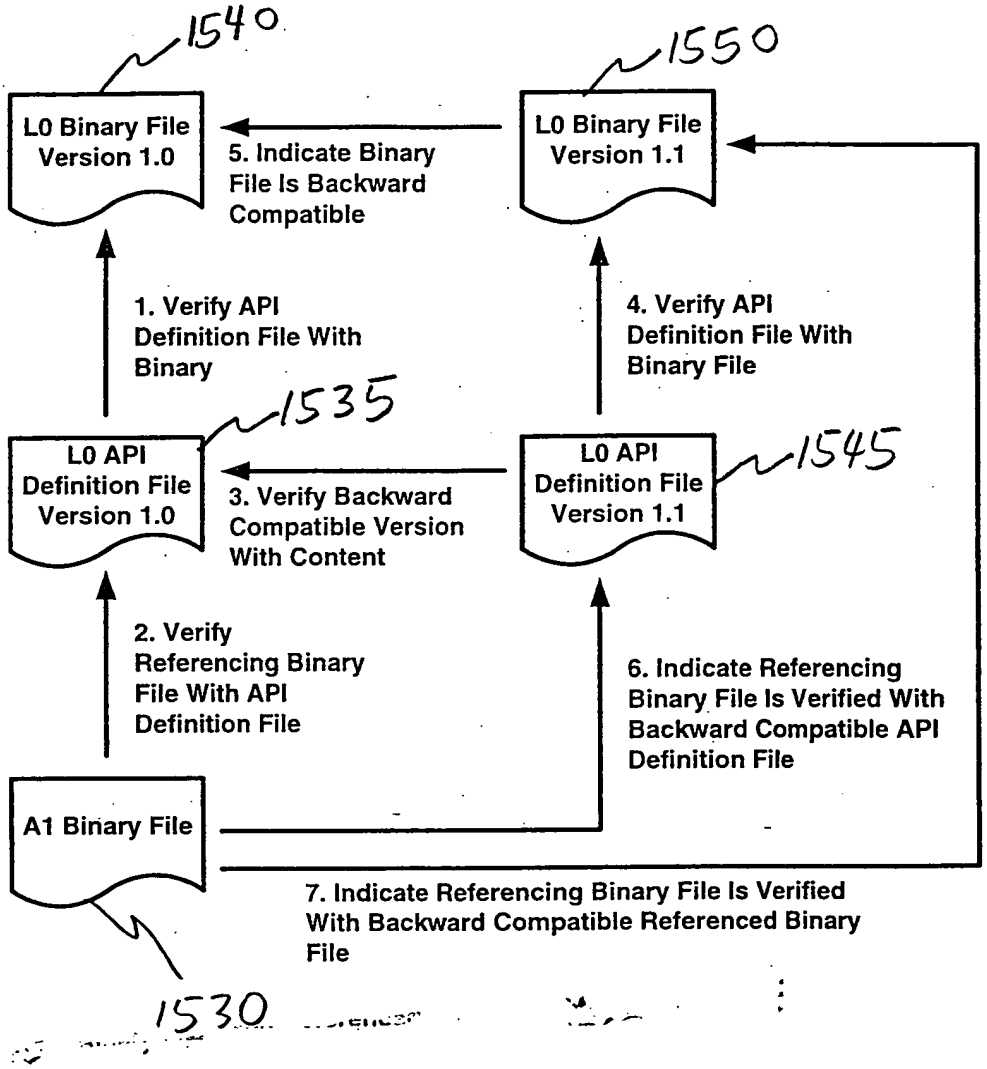
1540
1550
L0 Binary File Version 1.0
5. Indicate Binary File Is Backward Compatible
L0 Binary File Version 1.1
1. Verify API Definition File With Binary
4. Verify API Definition File With Binary File
1535
L0 API Definition File Version 1.0
3. Verify Backward Compatible Version With Content
L0 API Definition File Version 1.1
1545
2. Verify Referencing Binary File With API Definition File
6. Indicate Referencing Binary File Is Verified With Backward Compatible API Definition File
A1 Binary File
7. Indicate Referencing Binary File Is Verified With Backward Compatible Referenced Binary File
1530

Fig. 20A

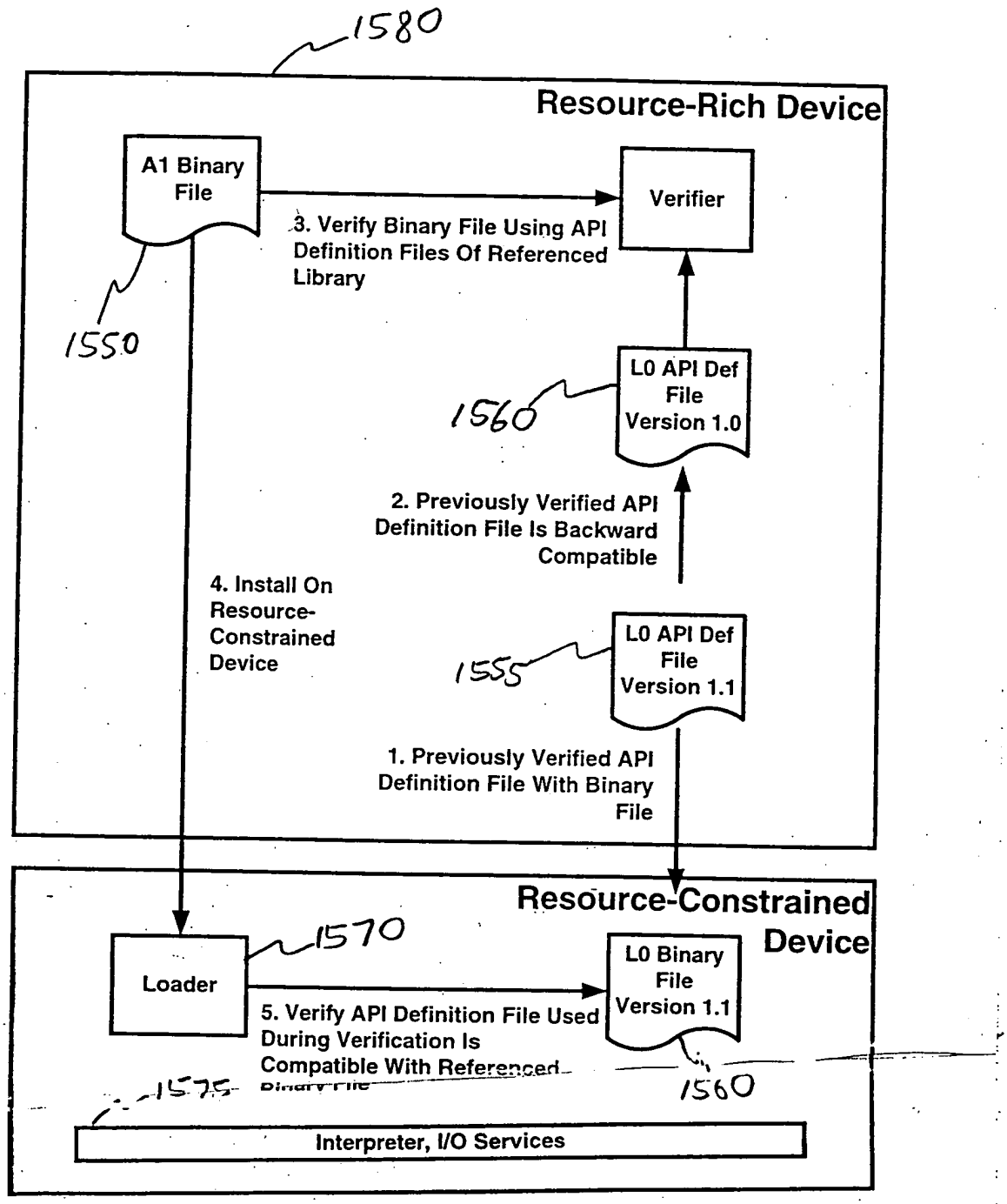
1580
Resource-Rich Device
A1 Binary File
1550
Verifier
3. Verify Binary File Using API Definition Files Of Referenced Library
1560
L0 API Def File Version 1.0
2. Previously Verified API Definition File Is Backward Compatible
4. Install On Resource-Constrained Device
1555
L0 API Def File Version 1.1
1. Previously Verified API Definition File With Binary File
Resource-Constrained Device
Loader
1570
L0 Binary File Version 1.1
5. Verify API Definition File Used During Verification Is Compatible With Referenced Binary File
1575
1560
Interpreter, I/O Services

Fig. 20B

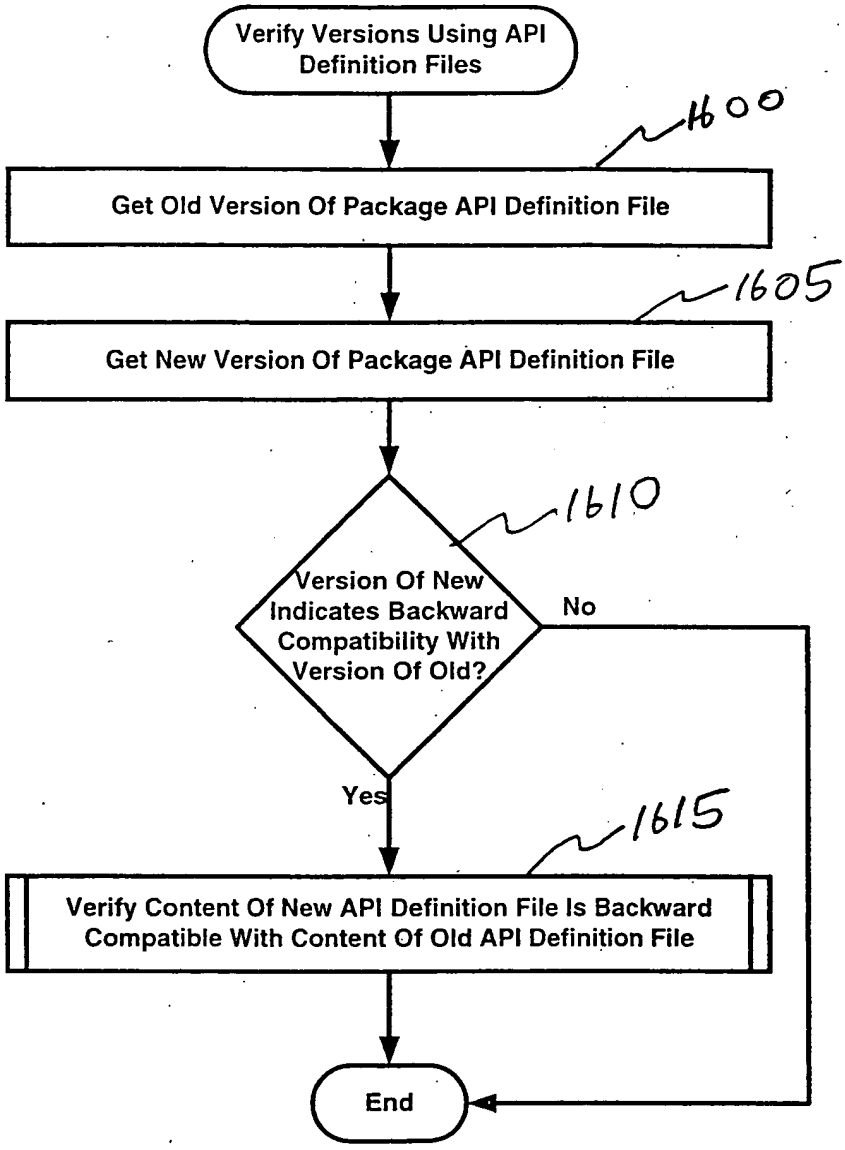
Verify Versions Using API Definition Files
1600
Get Old Version Of Package API Definition File
1605
Get New Version Of Package API Definition File
1610
Version Of New Indicates Backward Compatibility With Version Of Old?
No
Yes
1615
Verify Content Of New API Definition File Is Backward Compatible With Content Of Old API Definition File
End

Fig. 20C

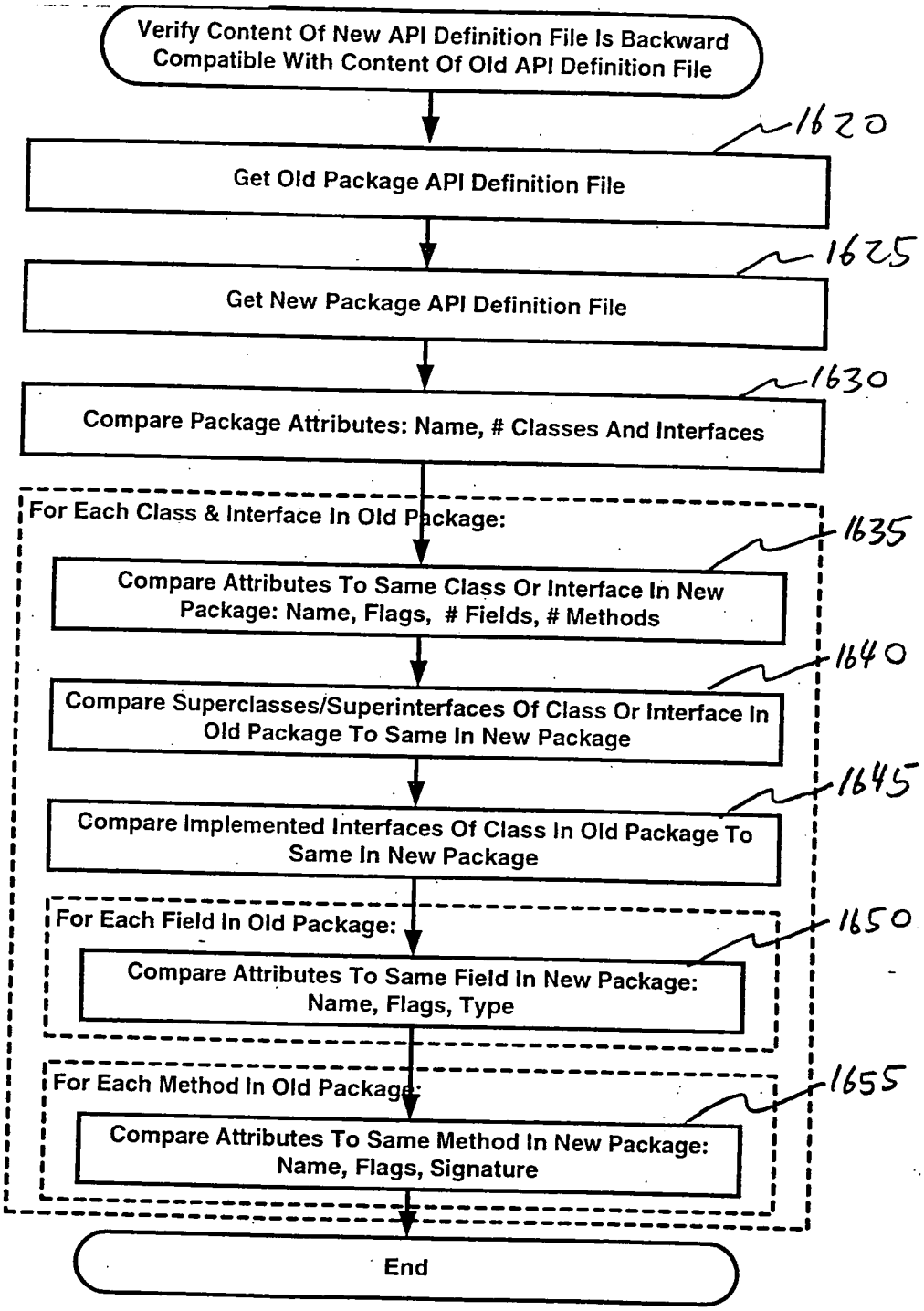
Verify Content Of New API Definition File Is Backward Compatible With Content Of Old API Definition File
1620
Get Old Package API Definition File
1625
Get New Package API Definition File
1630
Compare Package Attributes: Name, # Classes And Interfaces
For Each Class & Interface In Old Package:
1635
Compare Attributes To Same Class Or Interface In New Package: Name, Flags, # Fields, # Methods
1640
Compare Superclasses/Superinterfaces Of Class Or Interface In Old Package To Same In New Package
1645
Compare Implemented Interfaces Of Class In Old Package To Same In New Package
For Each Field In Old Package:
1650
Compare Attributes To Same Field In New Package: Name, Flags, Type
For Each Method In Old Package:
1655
Compare Attributes To Same Method In New Package: Name, Flags, Signature
End

Fig. 20D

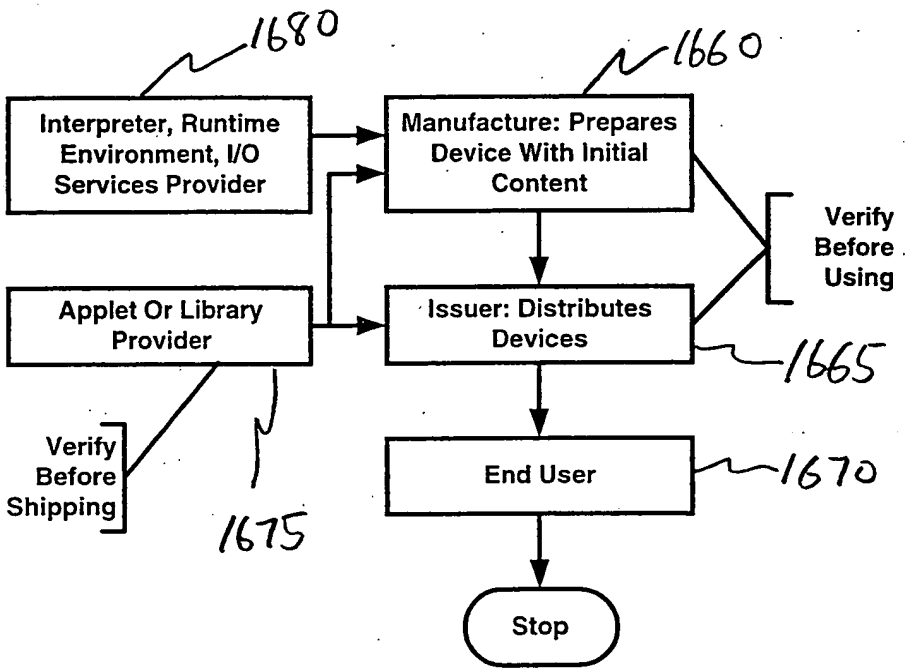
1680
1660
Interpreter, Runtime Environment, I/O Services Provider
Manufacture: Prepares Device With Initial Content
Verify Before Using
Applet Or Library Provider
Issuer: Distributes Devices
1665
Verify Before Shipping
1675
End User
1670
Stop

Fig. 21A

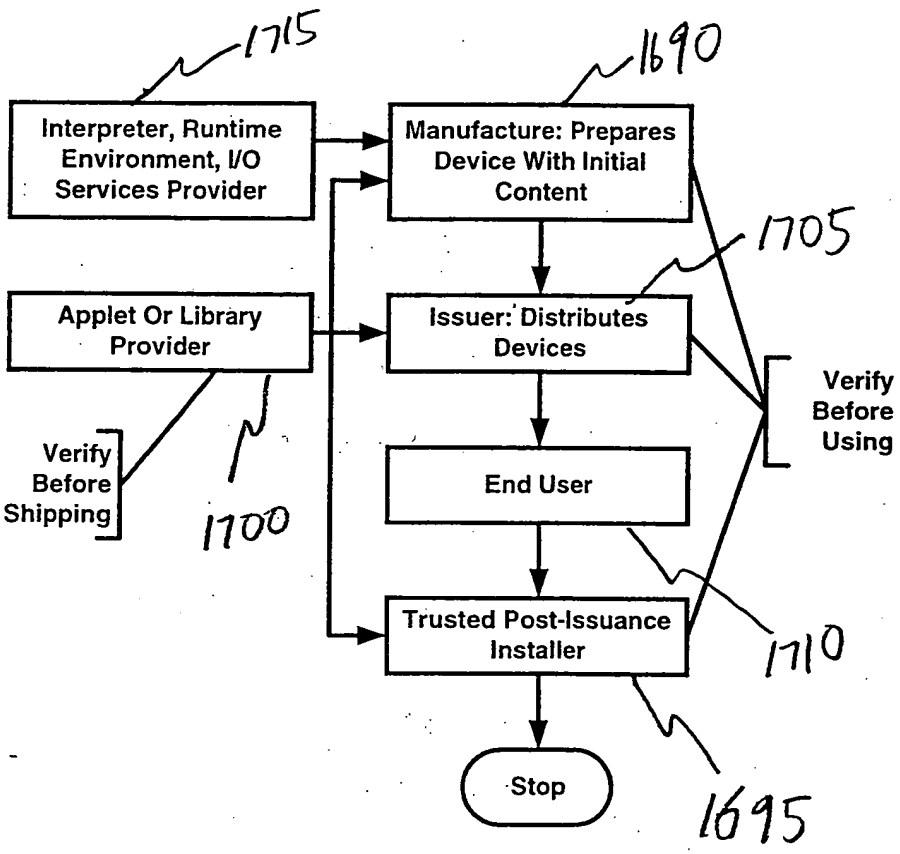
1715
1690
Interpreter, Runtime Environment, I/O Services Provider
Manufacture: Prepares Device With Initial Content
1705
Applet Or Library Provider
Issuer: Distributes Devices
Verify Before Using
Verify Before Shipping
1700
End User
Trusted Post-Issuance Installer
1710
Stop
1695

Fig. 21B

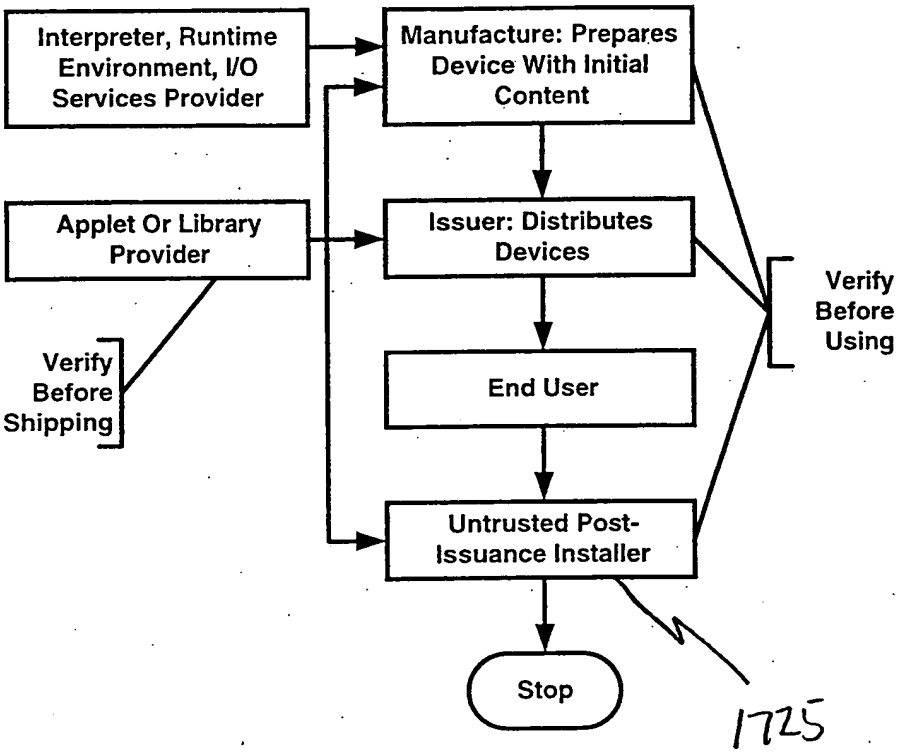
Interpreter, Runtime Environment, I/O Services Provider
Manufacture: Prepares Device With Initial Content
Applet Or Library Provider
Issuer: Distributes Devices
Verify Before Using
Verify Before Shipping
End User
Untrusted Post-Issuance Installer
Stop
1725

Fig. 21C

PRINT OF DRAWINGS
AS ORIGINALLY FILED

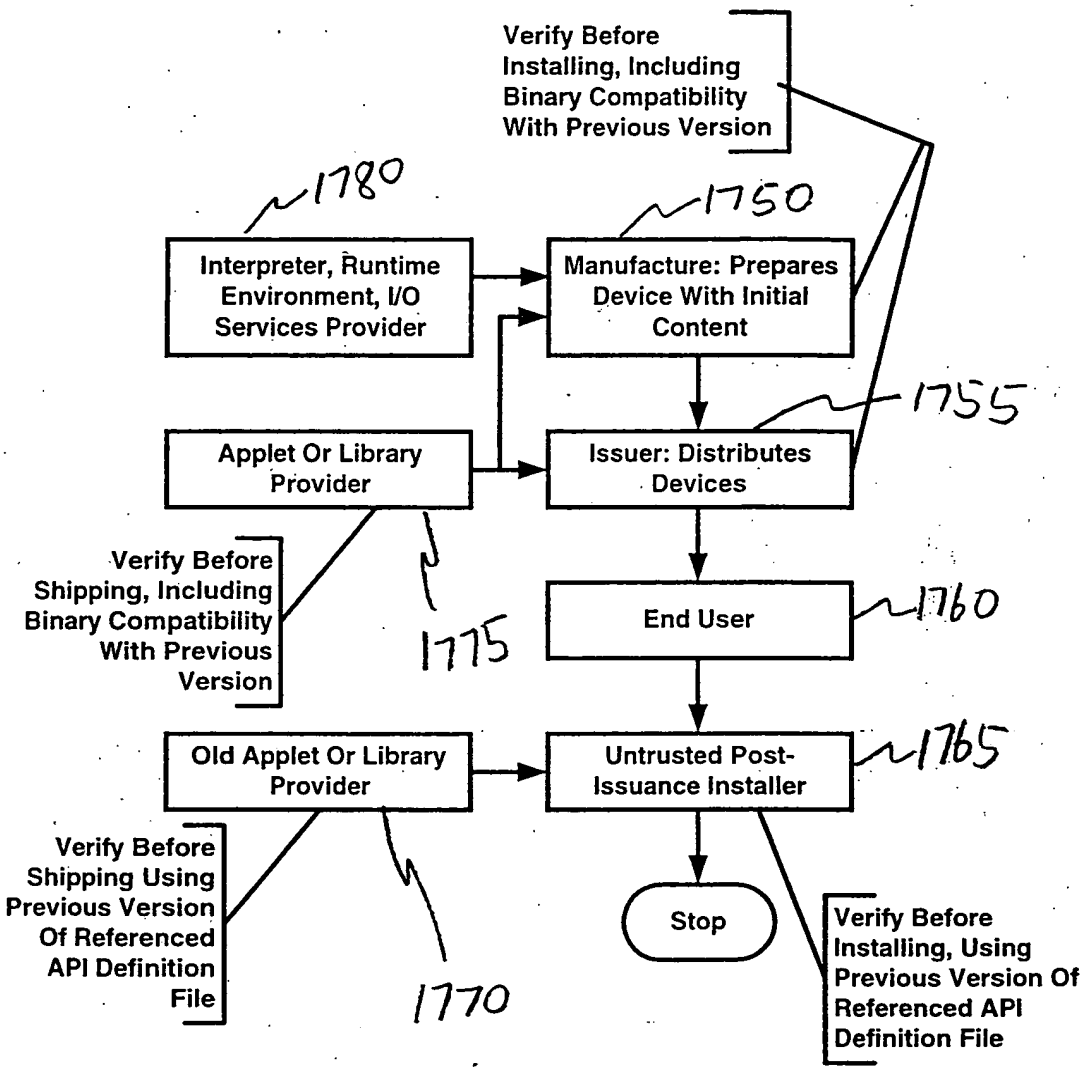

Fig. 21D